AEOS

Readers and Identifiers

Advanced information APxx03 / APx007

Third Party Readers and PIN

02 November 2012 Version 24

This information is furnished for guidance, and with no guarantee as to its accuracy or completeness; its publication conveys no licence under any patent or other right, nor does the publisher assume liability for any consequence of its use; specifications and availability of goods mentioned in it are subject to change without notice; it is not to be reproduced in any way, in whole or in part, without the written consent of the publisher.

0 CONTENTS

1. INTRODUCTION 4

2. GENERAL INFORMATION AP1003 / AP4x03 / AP6003 READERS 5
2.1. General connections at AP1003 5
2.2. General connections at AP4x03 5
2.3. General connections at AP6003 5
2.4. AEpack Firmware 6
2.5. AEpack Identifier settings 7
2.6. AP1003 / AP4x03 / AP6003 Data formats 7
2.7. AP1003 / AP4x03 / AP6003 Check identifier settings using AEpu logfile 8
2.8. Generic Identifier 9

3. AP1007 READERS 15
3.1. AP1007 Mifare classic card explanation 15
3.2. AP1007 Mifare Identifier and Firmware 15
3.3. Representation of CSN number 15
3.4. AP1007 / AP4x07 Special firmware 15

4. SETTINGS IN DATABASE 16

5. SELECTION AT FRONTEND 17
5.1. Leading zero's 17

6. PIN CODE - VERIFICATION 18
6.1. Overview available PIN-code devices 18
6.2. PIN-code terminals using Wiegand protocol 19
6.3. PIN-code terminal Commisioning (AEmon settings) 19
6.4. PIN-code terminal AEOS server settings 19
6.5. PIN-code sending a string as identification 19
6.6. Changing the number of digits for PIN-code (default = 4) 20
6.7. PIN-code string as decimal value 20
6.8. CodeGuard RS232 PIN-code terminal as verification 21
6.9. Other verification methods 23

7. AP1003 / AP4x03 / AP6003 protocols 24
7.1. RS232 protocols 24
7.2. Omron protocols 24
7.3. Wiegand protocols 27

8. AP1003 / AP4x03 firmware 32

9. Readers tested with AP1003 / AP4x03 / AP6003 33
9.1. General information 33
9.2. Deister PRX5, KPM 5/2, PRM5/2 Wiegand interface 34
9.3. HID EPIC 315 Wiegand interface 35
9.4. HID IQcard Mifare reader Wiegand interface 36
9.5. HID ProxPoint Plus 6005B Wiegand interface 37
9.6. HID HID ProPro 5355AGN Pin Wiegand interface 37
9.7. HID MiniProx readers 37
9.8. HID IClass RK40 PINPad 39
9.9. Legic PR74 RWC (LEG-SC-RW-002), Triple Eye 41
9.10. Omron 3S4YR-HNR-CMH-PCB Omron interface 42
9.11. Opticon NSR120 Barcode interface 43
9.12. INDALA FlexPass FPC-0500A+ 44
9.13. KeyProcessor SL-4 Magnetic OMRON T2 45
9.14. DALLAS IButton DS1971 + converter DS9097 46
9.15. Nedap Transit Reader 47
9.16. Datalogic DLS2032R-M0 Barcode RS232 interface 49
9.17. PIPS License plate recognition – RS232 50

9.18. Tattile Vega License plate recognition – RS232 51
9.19. Tattile Vega License plate recognition – IP 52
9.20. Sony XCI-NPR License plate recognition – RS232 53
9.21. Nedap ANPR License plate recognition – IP 54
9.22. Mobile Phone MC35i – RS232 interface 55
9.23. Storm PIN pad – RS232 interface 56
9.24. GEMPlus GEM Easy Access300 - Mifare 57
9.25. Dorset LID650 (Trovan) 58
9.26. IS3480 QuantumE (Metrologic) - Barcode 59
9.27. Convexs / Invexs readers (Nedap Mifare sector data NR0077 / NR0125) 60
9.28. IButton RS232 61
9.29. MOTOROLA – FlexPass FP3515A (Omron) 62
9.30. Satel CZ-EMM (EM) 63
9.31. PhG readers (e.g. Oris, Voxio) 64
9.32. PhG Crypt bus protocol 64
9.33. PhG Oris 100 Omron 66
9.34. PhG Aperio 200 Omron 67
9.35. PhG Voxio RS485 69
9.36. PhG Voxio 200 Wiegand 70
9.37. GigaTMS MF7 (Mifare) 71
9.38. Symbol LS9208 barcode reader (Barcode) 72
9.39. Evis EDI-05 RS485 (Legic, Mifare) 73

10. DOCUMENT HISTORY 74

1. INTRODUCTION

This user manual describes how to handle when connecting third party readers to AEOS, including connections, firmware, configuration using AEmon and settings at the server.

Third party readers can be connected to the AEOS system using the AP1003 / AP4x03, AP6003 the reader interface AEpacks.

In case Nedap Readers (e.g. Convexs or Invexs readers) are used, check their corresponding manuals. These Nedap Readers are capable of reading Nedap, Mifare and DESFire simultaneously.

Beside the readers mentioned in this document AEOS offers due to the used protocols the opportunity to connect almost every reader, as long as there is a suitable interface. Specially in combination with the Generic Identifier (see chapter 2.8) many possibilities are available.

Read **always** the corresponding manual of the AEpack and of the connected reader for more detailed information.

In any case the standard possibilities will not fit to your specific situation, contact the Nedap supplier for a customer specific solution.

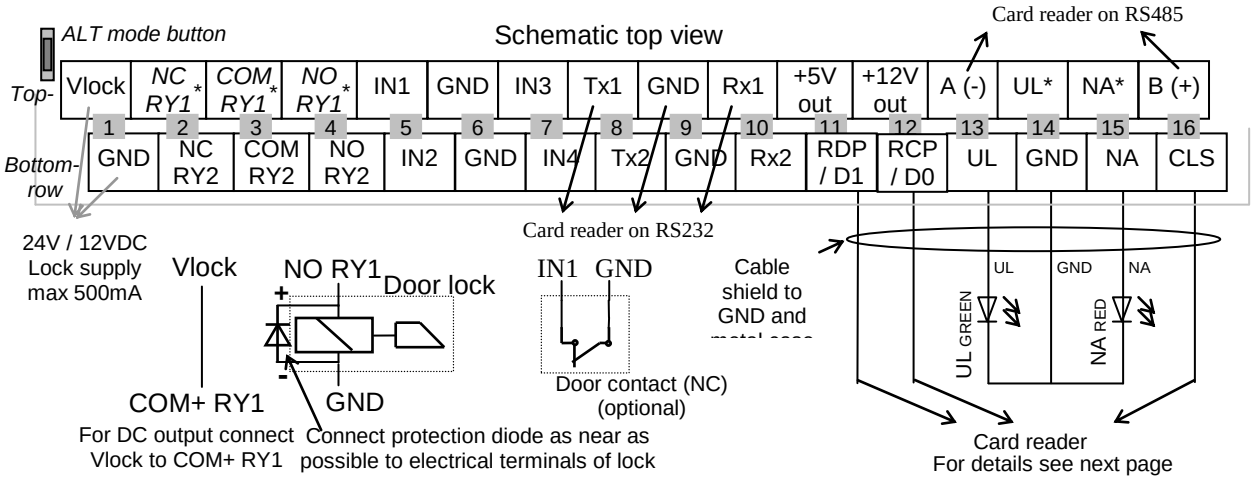
Remarks:

- Unfortunately this document will not be always 100% up to date. Check the latest available firmware for the wanted protocol.
- In most situations instead of the AP1003 also an AP4003 or AP6003 can be applied, check the availability of the firmware for this protocol.

2. GENERAL INFORMATION AP1003 / AP4x03 / AP6003 READERS

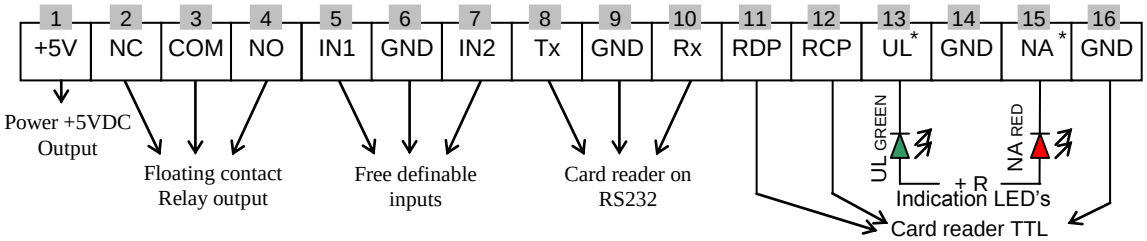
2.1. General connections at AP1003

See also the AP1003 Installation sheet. The card reader connections are made at the peripheral connector beneath the bottom cover of the AP1003.



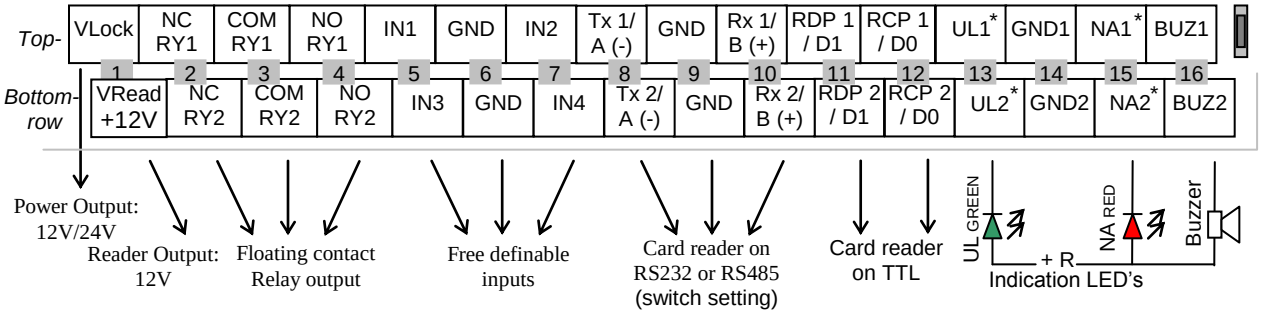
2.2. General connections at AP4x03

See also the AP4803 and AP4003 Installation sheet.



2.3. General connections at AP6003

See also the AP6003 Installation sheet.



2.3.1. Connections to readers

The TTL connection to the card reader is for TTL communication on connection points: RDP, RCP and CLS.

			Reader connection		
AP1003	**AP4x03**	**AP6003**	**Omron**	**Wiegand**	**Barcode**
11 (RDP)	11 (RDP)	11 (RDP)	/ RDP	/ Data 1	BC39
12 (RCP)	12 (RCP)	12 (RCP)	/ RCP	/ Data 0	
16 (CLS)			/ CLS		

For RS232 Rx1, GND en Tx1 are used.
RS485 (A, B) is also available.
For different card readers see table for correct connections. Detailed information can be found at following chapters.

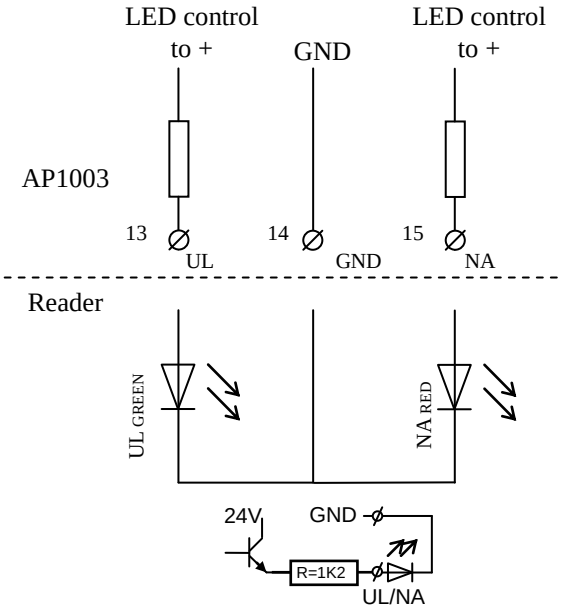
				Reader connection
AP1003		**AP4x03**	**AP6003**	**Reader**
8u (Tx1)	13u (A)	8 (Tx, A)	8 (Tx, A)	Rx (Receive data)
9u (GND)	9u (GND)	9 (GND)	9 (GND)	GND
10u (Rx1)	16u (B)	10 (Rx, B)	10 (Rx, B)	Tx (Transmit data)

The AP4x03 had no CLS connection, The AP1003 8u means point 8 on upper connector.

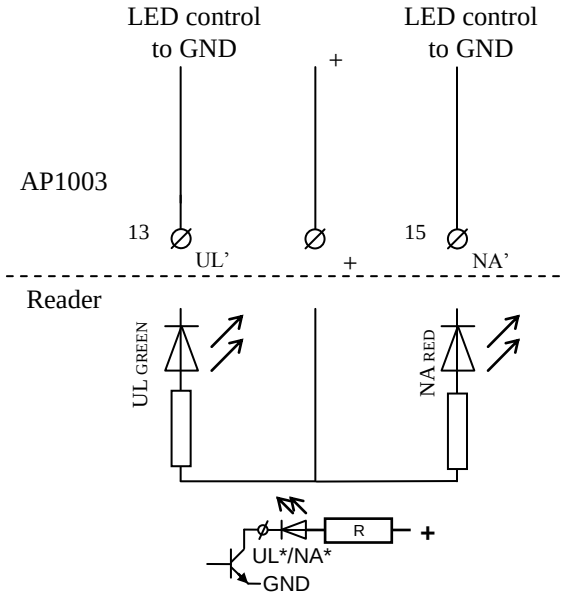
2.3.2. Connections to LED's

At connector points 13,14,15 of lower connector **common cathode** LED's can be connected, up from PMS B of AP1003 on points 13,14,15 of upper connector also **common anode** is possible (and AP4x03)

LED connections lower connector AP1003



LED connections upper connector (**AP1003** check hardware version) **AP4x03, AP603**



AP1003 Lower connector: ('Nedap standard'): UL – NA: 24V, R=1k2 on PCB, LED cathode to GND
AP1003 Upper connector and AP4x03: UL* - NA*: Open collector to GND. Resistor and Power must be supplied external.

2.4. AEpack Firmware

Depending of the chosen third party reader specific firmware must be loaded in the AP1003 / AP4x03, AP6003. See chapter 9 for the correct type of firmware needed.
Loading the firmware is done using AEmon:

- *View – Hardware*, select the corresponding AEpack
- Check the current *ApplicationFirmware* and the *BootFirmware* (versions must be correct), type of application firmware for this specific reader can be found in the table at chapter 9
- Load correct Firmware (can be obtained from CD or Internet), pro AEpack or for all same type of AEpacks simultaneously.

Remark: If new Application Firmware must be loaded check the version number, this must be suited to the version number of the Boot Firmware

2.5. AEpack Identifier settings

At AEmon the correct type and format of the card must be programmed:

- *View – Configuration*, select the corresponding AEbc (type Access Point);
- Go to tab *Properties* and select *Identifier Type* and if necessary identifier depending settings:
  - XS Badge
  - Wiegand Badge
  - Omron Badge (number of digits / Start at digit position)
  - Barcode Badge (Length / Start at position / Count in digits)
  - Mifare Badge
  - Generic identifier

These identifier settings must be fit to the identifier type stated at the AEserver at *Identifier Types*'. To check if these are corresponding, use AEmon, see also 2.7

2.6. AP1003 / AP4x03 / AP6003 Data formats

Depending of the used firmware at the reader several formats can be read. Each type of data has it's of format. This format can be checked at AEmon, using 'Events'.
E.g. AEmon shows: *Reader data: 030300E645C1B1*
Format = 0303, which represents Wiegand 37
This format is translated to the format that is sent to the AEpu (see also 2.7) for authorisation.

Table below only represents the first 2 digits of the data sent by the AEpack (the **reader data**).

Data type	Identifier
Nedap	
00 00	EF code
00 01	A code
00 02	EF (+ CB) code
00 03	B code
00 04	CF code
00 05	CF (+XF) code
00 06	DF code
00 07	DF (+XF) code
00 08	GF code
00 09	GF (+XF) code
00 0A	GF (+W) code
00 0B	GF (+W XF) code
00 0C	C code
00 0D	D code
00 0E	G code
Serial	
01 xx	Serial protocols
Omron	
02 01	Omron Track ISO 1
02 02	Omron Track ISO 2
02 03	Omron Track ISO 3
02 04	Omron max 200 ISO 2
Wiegand	
03 01	Wiegand 26
03 02	Wiegand 32
03 03	Wiegand 37
03 04	Wiegand 32Bin
03 05	Wiegand 44
03 06	Wiegand 35
03 07	Wiegand 64
03 08	Wiegand 128
03 09	Wiegand C1000

Data type	Identifier
Mifare	
05 01	Mifare CSN
05 02	Mifare Block data
05 03	Mifare DESfire CSN
05 04	Mifare DESfire Block data
05 05	Mifare Ultralight CSN
05 06	Mifare Ultralight Block data
05 07	
05 08	
05 09	Mifare Plus Block data
Legic	
07 01	Legic Prime CSN
07 02	Legic Prime Data
07 03	Legic Avant CSN
07 04	Legic Avant Data
HID	
08 01	IClass CSN
08 02	IClass Data
EM Marin	
09 01	EM4102
09 02	EM4050 Data

Remarks:

- From the first two digits, the first digit represents the type (e.g. Nedap or Omron or Mifare), the second the sub type (e.g. CF of DF(+XF) or CSN)
- Nedap uses different formats, this due to the different formats of e.g. the GF code that are made in the past
- For e.g. the CF code two formats are available: CF and CD (+XF). The XF part of the Nedap card represents that the second part of the card can be used e.g. for catering. This part has no influence to the identification part.
- The serial protocols are not been expanded, these are depending of the used hardware.

2.7. AP1003 / AP4x03 / AP6003 Check identifier settings using AEpu logfile

Following steps must be valid to check a card in the AEpu logfile:

- Card must be recognized at the used reader (LED on this reader blinks?)
- Card data must be transferred to the AEpack (LED's on AEpack at data signals RCP and RDP blinks?)
- ID LED on AEpack blinks: Data format is valid for AEpu
- Relay is activated: Card number is valid

In case of any problems check the AEpu logfile:

- At the AEpu logfile (*View – Logfile)* at presenting a badge a line will be added with the data as read by the AEpack. This can be as follows: (logging must be set on level 4=debug)

```
3 e 04.09.03 16:06:27:699 LogTextFile:= BadgeAccessEventEvent (1062684387698,
name=acc-portier, host=aepu01, id=1 {0x3,0x0,0x0,0x0,0x9d,0x6e}, direction=0)
```

Id=1	Id=1	Wiegand badge	(id=0: XS badge)
{0x3,0x0,0x0,0x0,0x9d,0x6e}	0x3	Series	
	0x0	Facility code	
	0x0	"	
	0x0,0x9d,0x6e	Cardnumber	

Facility code and cardnumber length depend of the type of identifier

- To compare the data read by the AEpack announce this badgenumber for a person on the AEserver. The data sent than to the AEpu can now be compared with the data as received by the AEpu from the AEpack. If these strings are identical, both identifier type settings are correct and the badge should be autorised.

Data sent from server to the AEpu (authorization command):

```
4 m 04.09.03 16:06:27:642 CarrierIdentityResolverImpl: assignment = carrier =
212, begin-validity = 04-Sep-2003 00:00 GMT+02:00, end-validity = 29-May-2149
00:00 GMT+02:00, identifiers = [{1 0x3,0x0,0x0,0x0,0x9d,0x6e}]
```

Data sent to the AEpu by AEpack (BadgeEvent):

```
3 e 04.09.03 16:06:27:699 LogTextFile: = BadgeAccessEventEvent(1062684387698,
name=acc-portier, host=aepu01, id=1 {0x3,0x0,0x0,0x0,0x9d,0x6e}, direction=0)
```

Data sent to the AEpu from the server is: **[{1 0x3,0x0,0x0,0x0,0x9d,0x6e}]**
Data sent to the AEpu from the AEpack is: **id=1 {0x3,0x0,0x0,0x0,0x9d,0x6e}**
Both strings must be identical, so the identifier settings for AEpack and AEserver are correct getting the authorisations at the AEpu.

2.8. Generic Identifier

The generic identifier type can be used to create your own identifier type and is not bound to a specific card type or card reader type. It contains a user definable sub type number which makes it possible to use up to 255 different generic identifiers within a single AEOS system. Besides the 'sub type' it contains properties which define how the access point should extract the badge number and customer code from the raw input badge data.
The generic identifier type transforms the data read from the reader ('raw data') to a free definable format. For receiving the card data from the connected reader the AEpack must have the correct reader interface and software: e.g. Wiegand32 or RS232.
The Generic identifier can also be used to make a filter on the XS customer codes or make an AEOS installation work only on selecting the XS customer codes. (See chapter 2.8.6)

2.8.1. Generic Identifier before 2.0.4 and after 2.0.4

With AEOS version 2.0.4 the Generic Type is been improved, and also some fields at AEmon have become different names. The Format types have been changed, see table below:

AEOS > 2.04				AEOS < 2.0.4		
Type	**Example**	**Representation**		**Type**	**Hex Representation**	
Ascii Alphanummeric	30-80 hex (31='1')	-	-	Alpanummeric	-	-
Ascii Hexadecimal	30 – 3F	Hex	Numeric	Hexadecimal	Yes	No
Hexadecimal	0 – F (real hex)	Hex	Numeric	Binary	Yes	No
BCD	0 – 9	-	-	BCD	-	-

The *Representation* is the way the data is entered and represented at the front end of AEOS. If e.g. *Hex* is chosen, the data read from the reader must be entered *hexadecimal* (eg. E46C). If *Numeric* is chosen, the data is been translated to *Numeric* (58476), and must also be entered at the AEOS front end.

BCD: Binary Code Decimal, bytes are paired: string '123456'is split up into bytes as'12', '34', '56'.

2.8.2. Generic Identifier type properties

The settings made in AEmon must be matching to the settings made in the AEOS server for the identifier type. For this purpose the settings for the server are automatically shown in AEmon.

- **Sub type**
  A number in the range 0..255 which makes a generic identifier unique. Number must be unique for those AEpu's using this generic type, at the server this number must be equal.
  By using different sub types several generic identifier types can be defined into the system.
  By using the same number at different AEpacks (access points) with different settings you can use different readers but at the AEOS front-end only one identifier has to be issued.

- **Badge number data format**
  Format or the badge number part in the raw card data, the following formats are supported: (ID is the badge number that is entered at AEOS)
  - *Ascii Alphanumeric*, the data consists of a number of ascii characters which forms the badge identifier.
    Raw data: 31 32 41 4C (4 ascii bytes)
    ID: 12AL
  - *Ascii Hexadecimal*, the data consists of a number of ascii characters which represents binary data, an additional property *Representation* (Hex or Numeric) defines if the binary data should be interpreted as a decimal number or as hexadecimal data (E.g. data is received as '012AD', but must be shown as '4781' at AEOS, than this additional property must be disabled: data is NOT represented as hexadecimal)
    Raw data: 31 32 41 44 (4 ascii bytes: 30 to 39 and 41 to 46)
    ID: 12AD Hex representation)
    D: 4781 (Numeric representation)

- *Hexadecimal*, the data consists of a number of bits which forms the badge identifier. It has also the additional 'Hex representation' property
  Raw data: 1B3C (4 hex bytes: 0 to F)
  ID: 1B3C (Hex representation)
  ID: 6972 (Numeric representation)
- *BCD*, the data consists of a number of Binary Coded Decimal nibbles.
  Raw data: 2475 (4 bcd bytes: 0 to 9)
  ID: 2475
  BCD data takes one (hex) byte for every two nibbles (so 24 is one byte and 75 is one byte)
- *Ascii Numeric*, (up from AEOS 2.4.3) the data consists of a number of ascii numeric characters which forms the badge identifier. It has also the additional 'Hex representation' property, which converts the decimal number to Hexadecimal.
  Raw data: 31 32 33 34 (4 ascii bytes: 30 to 39)
  ID: 4D2 Hex representation of 1234)
  ID: 1234 (Numeric representation)

- **Representation**
  The representation type is the way the ID must be entered and represented at the AEOS user interface. This means that e.g. a Hexadecimal number (1B3C) is translated to Numeric (6972) and 6972 must be entered for e.g. Announcing an employee with this cardnumber

- **Badge number data bit position**
  The offset of the data in the raw card data,
  0 = 1st bit, 1 = 2nd bit
  Remark: The first 4 bytes from the data from AEpack: (eg 0x0101 000115) is an internal code and **not** a part of the raw card data (in this case 000115), also for an XS badge: 0x0005 15A00115 the raw data is: 15A00115.
  The raw data can be checked at AEmon using Events or the Eventlog at debug level 4.

- **Badge number data length**
  Length of the input data in bits(Binary), digits(Bcd) or bytes(Alphanumeric, Hexadecimal) .
  When length = 0 **all** data (starting at the data bit position) is taken as input data.

- **Customer code data Fixed code**
  If checked a fixed customer code is placed for the Badge number.

- **Customer code data format**
  See Badge number data format.

- **Customer code data bit position**
  See Badge number data bit position.

- **Customer code data length**
  See Badge number data length.
  When length = 0 **no** customer code data is read.

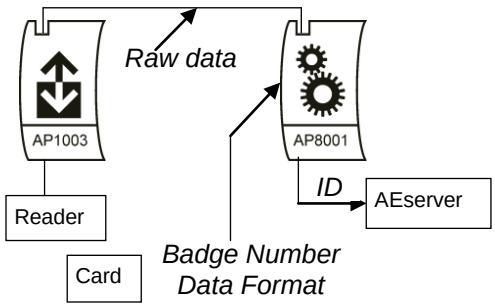
- **Target identifier type**
  With AEOS version 2.3.1 the option *Target identifier type* is added with the Generic Identifier Type. With this option you can **transform** the Generic Identifier Type to one of the know *Identifier types* in AEOS, using the Generic *Sub type* as link to the Identifier type. At AEmon the *AEserver settings* are now greyed out (are not to be used), at AEOS you must select the transformed type and not the Generic Identifier. (See AEbc AccessPoint descriptions for detailed information)
  Up from version 2.4 a check can be applied on the identifier type.

- **Data type filter**
  With AEOS version 2.4 the option *Data type filter* is added with the Generic Identifier Type. With this option you can **check** on the first two data bytes, presenting the identifier type.

2.8.3. Generic Identifier type examples

The Badge Number Data Format is the data that is read by the AEpack (e.g. AP1003) and transferred to the AEpu.

The reader sends the card data in e.g. Wiegand32 format to the AP1003 (AP1003 must contain correct firmware to be able to interface to the reader). The AEpu converts the raw data using the Generic Identifier Type to the ID number and uses this for authorisations (and sends this to the AEserver).



Example 1: **Ascii Hexadecimal, Numeric representation**
Ascii – hexadecimal characters '6CF3' representing the decimal number '27891', no customer code.
Raw card data is (in hex): 0x36434633 (36 ='6', 43='C', 46='F', 33='3')
AccessPoint settings: Generic Identifier Sub type: 1

Badge number			**Customer code (none)**	
Format	Ascii Hexadecimal	***Numeric***		
Bit position	0			
Length	4 (bytes)			

Raw data: 0x36434633
ID: 27891 (Numeric representation)

Example 2: **Ascii Hexadecimal, Hexadecimal representation**
Same as example 1 but badge is hexadecimal represented as '6CF3', no customer code.
Raw card data is (in hex): 0x36434633 (36 ='6', 43='C', 46='F', 33='3')
AccessPoint settings: Generic Identifier Sub type: 2

Badge number			**Customer code (none)**	
Format	Ascii Hexadecimal	***Hex***		
Bit position	0			
Length	4 (bytes)			

Raw data: 0x36434633
ID: 6CF3 (Hex representation)

Example 3: **Hexadecimal, Numeric representation**
Same as example 1 but raw badge data is binary, no customer code.
Raw card data is (in hex): 0x6CF3
AccessPoint settings: Generic Identifier Sub type: 3

Badge number			**Customer code (none)**	
Format	Hexadecimal	***Numeric***		
Bit position	0			
Length	16 (bits)			

Length: 4 nibbles (6CF3) of 4 bits -> 16 bits

Raw data: 0x6CF3
ID: 27891 (Numeric representation)

Example 4: **BCD data**
Same number '27891' but raw badge data is bcd , no customer code.
Raw card data is (in hex): 0x027891
AccessPoint settings: Generic Identifier Sub type: 4

Badge number			**Customer code (none)**	
Format	BCD			
Bit position	4			
Length	5 (digits)			

Bit position=4: the first nibble (0) is skipped -> badge number starts at '2'

Raw data: 0x27891
ID: 27891

Example 5: **Nedap CF badge**
Nedap CF-badge processed as Generic Identifier Type
The data received at the badge input is (in hex):
0x000515A00059
The first two bytes (00 05) are not a part of the card data and should be skipped so the raw card data is:
0x15A00059
The first three nibbles form the binary customer code (15A), the other nibbles form the BCD code badge number (00059)
AccessPoint settings: Generic Identifier Sub type: 5

Badge number			**Customer code**		
Format	BCD		***Format***	Hexadecimal	***Hex***
Bit position	12 (bits)		***Bit position***	0	
Length	5 (digits)		***Length***	12 (bits)	

Raw data: 0x15A00059
ID: 59

2.8.4. Generic Identifier type advanced use

An advanced example of the Generic Identifier Type is e.g. to use AP1003 connected to an XS070 and AP1001 readers in one system. At AEOS only one identifier must be used for this (otherwise for each employee two identifiers must be used), so the Generic Identifier Sub Type number must be equal for all situations (in this example: 6)

XS070: Is connected to AP1003 using RS232 protocol, customer code is checked at the XS070, but must be added as fixed data to make it identical to the AP1001 data.
XS070 sends data serial as N000345 <CR><LF>. The firmware at the AP1003 (AP1003_rs232) converts this data to the raw data: 0x000345 (the data shown in AEmon is 0x101000345))
The data received at the badge input is (in hex):

0x0101000345

The first two bytes (01 01) are not a part of the card data and should be skipped, the raw card data is:

0x000345

AccessPoint settings: Generic Identifier Sub type: 6

Badge number			**Customer code Fixed code = yes**	
Format	BCD		***Code***	15A
Bit position	4		***Format***	Hexadecimal
Length	5 (digits)		***Length***	2

ID: 00345
Customer code: 15A, fixed into two hex bytes (01 – 5a)

AP1001: Nedap CF-badge processed as Generic Identifier Type
The data received at the badge input is (in hex):

0x000515A00345

The first two bytes (00 05) are not a part of the card data and should be skipped so the raw card data is:

0x15A00345

The first three nibbles form the binary customer code (15A), the other nibbles form the BCD code badge number (00345)

AccessPoint settings: Generic Identifier Sub type: 6

Badge number				**Customer code**		
Format	BCD	***Hex***	No	***Format***	Hexadecimal	***Hex***
Bit position	12 (bits)			***Bit position***	0	
Length	5 (digits)			***Length***	12 (bits)	

ID: 00345
Customer code: 15A

For both the AP1003 and the AP1001 the used Generic Identifier Type settings should result in the same setting at the AEOS front end (*Identifier Types*)
AEOS Settings for Generic Identifier Sub type 6:

Printable	No	***Print receipt***	
Identifier length	3		
Identification Type	BCD		
Identifier min. value	1	***Identifier max. value***	99999
Has different intern.	No	***Add leading zeroes***	No
Customer code length	2		
Customer code	15A		
Subtype	6	***Customer code type***	Hexadecimal

2.8.5. Generic Identifier type problem solving

If making the correct settings for the Generic Identifier type causes problems, so the cards can not be authorized try following steps:

- ID led of AEpack must blink, otherwise received data is not been understood by AEpack
- Clear logfile in AEmon
- Set logfile level of AEmon to debug (level 4)
- Announce a card from this identifier type
- Offer this card to the external reader (or antenne)
- Get the logfile with above data
- Compare both data strings (data that is sent to the AEpu from the server with the data that is sent by the AEpack to the AEpu)
- If these strings are identical, the correct settings are made.

See for more detailed information chapter: 2.7

2.8.6. Generic Identifier checking and filtering on XS customer codes

Another advanced example is to make a filter, for preventing that not all customer codes are shown in the AEOS Event monitor. This is just only an example; there are many other situations where using the Generic Identifier can help solving some problems.

Situation: Customer uses TransIT readers with Boosters and 3 different XS Customer codes. However at this Transit reads also badges from other customers (it is placed at a public road).
Solution: Add an additional accesspoint in which the Generic Identifier type together with the authorized badges make a selection of only those Customer codes which may be valid. The *Badge output* of this Filter accesspoint is linked to the *Badge input*. Now only the data from the valid (authorized) badges from the Filter accesspoint is read by this second accesspoint.

Keep in mind that in some situations duplications of this used customer codes can occur, because the type of badge read cannot be checked: you cannot see the difference between a C010 and a CF010 badge. C010 is seen as 000C15A 00001 and CF010 as: 000515A 00001.
Remark: As the first 4 bytes cannot be used no difference between the C and CF (and D and DF) customer code with the same number. Up from AEOS 2.4 a check can be made on the first 4 bytes.

2.8.7. Generic Identifier checking advanced setting

- Up from 2.3.x it is possible to change (*Target identifier type*) the identifier from Generic to one of the other available identifier types. This enables easier integration in existing systems
- Up from 2.4 a check can be applied on the first two bytes of the received data string (*Data type filter)*. This enables you to distinguish between e.g. Wiegand26 and Wiegand37.

3. AP1007 READERS

AP1007 readers are used to read Mifare cards.

3.1. AP1007 Mifare classic card explanation

A default Mifare classic card has 16 sectors. Each sector has 4 blocks of 16 bytes each. The last block of each sector contains the special data (security keys and access bits)

Block 0: 16 bytes		
Block 1: 16 bytes		
Block 2: 16 bytes		
Key A	Access bits	Key B

Sector 0, block 0 (the first 16 bytes on the card) is special, it contains the Mifare serial number (CSN=CardSerialNumber), this is a ReadOnly block.

Key A is needed to read or write the first 3 blocks of data of this sector.

3.2. AP1007 Mifare Identifier and Firmware

For reading Mifare cards the information on the card that must be used as the identifier number can differ, depending on the location where this information is placed on the card

- Identifier is the Card Serial Number (CSN).
  The CSN is a 32 bit unique number, no keys are required to read this number, no authentication is available.
  Using the CSN number as identifier requires the default firmware: (AP1007_543v200.mhx)
  Card data is presented at AEmon as 0501 AABBCCDD (AABBCCDD = CSN number in Hex)
- Identifier is stored in one of the sectors
  Identifier has no standard format, and is protected by keys and authentication.
  This data can only be read with the correct keys.
  Card data is presented at AEmon as 0502 5566778899 (5566778899 = sector data data)
- Identifier is the Card Serial Number (CSN) combined with information stored in one of the sectors
  Identifier has no standard format, and is protected by keys and authentication.
  This data can only be read with the correct keys.

To prepare an AP1007 with the correct firmware Nedap needs:

- Position of the 'data ID'
  - Sector number
  - Block number
  - Byte offset
  - Number of bytes
- Key of the concerning sector

With above information a customer depending firmware can be made and will be named e.g. as follows: AP1007s02_543v201.mhx

3.3. Representation of CSN number

Depending of the card manufacturer and the reader manufacturer the way the CSN number is represented can differ, however they intend to be the same number.
Example: Assume CSN number is: E29CB81E (E2 9C B8 1E). Depending of the used reader this number can be represented as:
E2 9C B8 1E or as 1E B8 9C E2
To avoid problems in AEOS, Nedap offers firmware for the AEpacks to handle both representations for the AP1003 / AP4003 readers. (Wieg32bin / Wieg32binRev). The Nedap Mifare readers present the data as normal (so not Reversed).

3.4. AP1007 / AP4x07 Special firmware

For the AP1007 / AP4007 special firmware is available if data from one of the sectors must be read. See the special documentation for these specials. Beside special firmware up form medio 2009 it is possible for some AEpacks to load a AEpack configuration file on the standard firmware for reading the customer specific settings (this file can be generated by AEreco).

4. SETTINGS IN DATABASE

Up from AEOS version 1.5 the identifiers can be added using the web-based user interface. At versions before 1.5 the identifiers must be added directly into the database (Table: identifiertype). See table below for the different cardtypes and their according data that must be added into the database:

Name	Class	Series	Facilitycode	Customercode
Barcode	4			
Omron	3			
Mifare default	5	1		
Mifare (sector)	5	2		
Wiegand-26	2	1		
Wiegand-32	2	2		
Wiegand-37	2	3	<space>	<space>
XS Badge	1	5		15a
Generic Identifier	6	<subtype>		

Remarks:

- The string filled in column **Name** is also represented at the frontend, this name can be freely defined by the user (attention: in some cases this name must be fixed)
- **Mifare default (CSN)** reads the Card Serial Number of the Mifare card
- **Mifare (sector)** reads data out of one of the sectors (secured)
- **Facilitycode** is depending of the card data. The facilitycode can be retrieved from the card data, check the string from the logfile.
  If the Facilitycode and custumercode for Wiegand 37 are not use, a space must be given here.
- **Customercode** only for Nedap badges, must be retrieved from Nedap Groenlo, depending of the type of card (C or CF, DF, etc) also the Series must be changed (check Nedap Groenlo).
- **Wiegand types:** Following Wiegand types are mapped on Wiegand 37 (so on the AEserver these types are presented as Wiegand 37):
  Wiegand 35, Wiegand 36, Wiegand 37. Wiegand 44
- **XS Badge:** Following subtypes (series) are used for the XS Badges (these are different from the data that is send form the AEpacks , see chapter 2.6):

XS Badge	Series	XS Badge	Series
		A – code	0
		B – code	1
CF – code	5	C – code	2
DF – code	6	D – code	3
GF - code	7	G – code	4

Remark: The settings above are the database settings, for the data that the AEpack is sending, see chapter 2.6 (AP1003/AP4x03/AP6003 Data formats).

5. SELECTION AT FRONTEND

At the user interface of AEOS up from version 1.5 the identifier types can be selected to been added to the database. The name of the identifier on the screen is equal as the field ‘Name’; in the above table. This name can in most of the cases be freely chosen (in some cases this name can also be used for import / export functions, and may not be changed!)
The setting at the identifier types determines in which format the card number must be entered.

Identifier min/max value: Depeding of the selected type of identifier, the minimum and maximum value possible for this selected format are already filled in. If needed these settings can be changed.
Only those fields who are mandatory must be filled in, the other fields are optional.

Remark: For the Generic Identifier type use the settings as been shown at AEmon when the settings for the AccessPoint is been made (right part of the screen)

5.1. Leading zero’s

Leading zero´s are automaticly added.

6. PIN CODE - VERIFICATION

There are several types of PIN code possible: See table below (6.1).
Also any terminal sending a readable string to AP1003 can be used as identification.

Remark: AEOS version < 2.1: If a PIN-code is 3 times incorrect entered the person will automatically be blocked by the AEOS server for **all** doors.
With AEOS 2.1 the blocking of the person at entering a wrong PIN-code can be arranged by using the *Action on Events*. (If no Action is defined, there is no blocking of this person.)

6.1. Overview available PIN-code devices

AEOS < 2.1 see table below (AEOS >2.1 see remark 5 and 6)

Device	Card	AEpack firmware	AEOS Identifier	Remarks
Invexs M..K..170	Mifare/Nedap	APx003rs485NR(2)	XS / Mifare	
CodeGuard	-	All with serial hardware component	-	PIN only
Storm WE1KT1	-	AP1003wiegs11 No AP4003 firmware	Wiegand26	PIN only
Storm PIN Pad	-	AP1003rs232Storm No AP4003 firmware	Generic	Identifier
Storm PIN Pad	-	All with serial hardware component	-	PIN
Deister KPM5	Mifare CSN	AP1003wieg37deisterKPM5 AP1003wieg37KPM5p1 AP1003wieg37KPM5p2 No AP4003 firmware	Wiegand37	
HID ProxPro	Prox	AP1003wieg26 AP4003wieg26	Wiegand26	**A**
HID iCLASS RK40	iCLASS	AP1003wieg26 AP4003wieg26	Wiegand26	**A**
HID iCLASS RK40	Mifare CSN	AP1003wieg32bin AP4003wieg32bin	Wiegand32Bin	**A,B**
HID iCLASS RK40	Mifare CSN	AP1003wieg32binRev AP4003wieg32binRev	Wiegand32Bin	**A,C**
PhG Voxio	Several	AP1003rs485PhGCrypt AP4003rs485PhGCrypt	Generic	

Remarks:

1 CodeGuard can be connected to those AEpacks with the serial hardware component available (check PMS of this AEpack for more information)
2 HID iCLASS readers need to have corresponding options: (check also the HID information)
  A: iCLASS PIN-options 00, 09 and 11:
    - 00: buffer one key, no parity, 4 bits message
    - 09: buffer one key, add complement, 8 bits message (Dorado format)
    - 11: buffer one key and add parity, 5 bits message
    - 2 times same key no parity, 8 bits message

    All these four options are automatically read by AEOS, and interpreted as PIN data.
  B: iCLASS CSN-option 1
  C: iCLASS CSN-option 0
3 Readers without remark 'PIN only' must have separate reader, the devices represented above generate only the PIN-code.
4 Storm PIN code reader only can deliver PIN-codes with 4 characters.
5 Up from 4-2007: The standard Wiegand Protocols and most specials can all handle the 3 different PIN-code options (4, 5 and 8 bits code) for AEpack versions v2. For *Wieg128maxMultiPin* firmware the PIN code is sent as one complete string.
6 AEOS versions **up from 2.1**: Almost all protocols are available for the AP1003 and the AP4003 (for AEpack versions v2).
7 Leading zeros are skipped when using the PIN code. This means that a e.g. 6 digit PIN code of 000123 acts identical to PIN code 123 (and as 0123).
8 PhG Voxio readers connected by the PhG Crypt protocol can be used with the PIN code of AEOS (fixed 4 digits)

6.2. PIN-code terminals using Wiegand protocol

Up from 4-2007 all standard Wiegand protocols and most specials can be used with PIN-code.
Some PIN-code terminals (e.g. Deister KPM5, see table above) uses the Wiegand protocol for transmitting the entered PIN-code. For connections see chapter 9
These PIN code terminals can only be used as verification, first a badge has to be presented, than the PIN code terminal askes for the correct PIN code.
As the Wiegand PIN-code terminal uses Wiegand for communication no communication settings has to be made.

6.3. PIN-code terminal Commisioning (AEmon settings)

For selecting PIN-code following must be selected at AEmon at the AccessPoint:

- *View – Configuration*, select the corresponding AEbc (type Access Point);
- Go to Tab *Properties*, select *Verifier type*, and select the correct type of verifier (e.g. for Wiegand PIN-code terminals: e.g. PIN-port)
  At *Properties* the Timeouts can be set:
  - Digit Timeout: maximum time between entering the different digits
  - Start Timeout: maximum time before first digit must be entered

  And the way an Alarm Ouput is activated (must be configured at Outputs) in case of emergency or threat (last digit is entered wrong of 3 times).
  If serial communication is used, also the communication settings must be made.
- At *Inputs*: Select the *Verification PINport* for Wiegand terminal of *Verification SerialPort* for the Code-Guard
- Using PINport you can (up from AEOS 2.4.3) determine to enter only the PIN and not the end-character.
- With AEOS 2.4.3 leading zeros must be entered, so PIN *12* in a 4 digit PIN system must be entered as *0012*.

6.4. PIN-code terminal AEOS server settings

To use PIN code at the server following must be done:

- License option *Verification* must be available
- At the table 'system properties' in the database the length of the used PIN-code must be set (default 4 digits).
  Attention: This value may only be changed if no PIN-codes are already used.
- The PIN code option must be available (at AEOS server version before 2.0 activate the option PIN code by removing (or rename) the file 'PINlock')
- In the Announce persons screens now it's possible to add the PIN-code
- Pro AccessPoint the schedule of not using the PIN-code can be defined

6.5. PIN-code sending a string as identification

Any terminal sending a string which can be identified by the AP1003 can be used for identification.
This received string is handled as the card number, so this is not the real PIN code option.
Please check if the communication format of the selected PIN-code terminal can be read by one of the AP1003 applications (check firmware).
Best way is to use the translation table, otherwise the (secret) PIN-code will be directly visible in the eventmonitor.
For using a Storm PIN code terminal (Nedap art. Nr: 9896090) special firmware is available to read the data (as identifier). This firmware handles the received data as one string until a # is been received, and starts again if during 5 seconds no data is been received (see chapter 9.23). For other type of terminals this firmware can also be used. Check the appropriate chapter for more information.

6.5.1. PIN-code as identification AEOS server settings

At AEOS make a corresponding identifier and assign this one to a person. (Of course it's also possible to make one general code for a group of persons.)
The length of the PIN-code is depending of the type of terminal used, see for this part the AP1003 settings.

6.6. Changing the number of digits for PIN-code (default = 4)

Default the number of digits used is 4 (can be checked at AEOS System Properties). This can be changed, only for **ALL** PIN-codes in the system:

- Stop AEOS services
- Change in table *Systemproperty* the value *Pincodelength* from 4 to e.g. 5
- Delete **all** records in table *Pincode* and table *Verificationassignment*
- Start AEOS services
- Reload **all** entrances
- Reassign again all PIN codes to all employee/visitors (they all have been deleted before)

Remark: Consult Nedap first before changing the number of PIN-digits.

6.7. PIN-code string as decimal value

The PIN code can also be used to control outputs.
This can be achieved with the PINtoDecimal AEbc. With version 2.3.10 this object can be triggered to activate the PINcode terminal. Detailed information can be found in the AEbc Miscellaneous descriptions.

6.8. CodeGuard RS232 PIN-code terminal as verification

The CodeGuard PIN code terminal can only be used as verification, first a badge has to be presented, than the PIN code terminal askes for the correct PIN code.

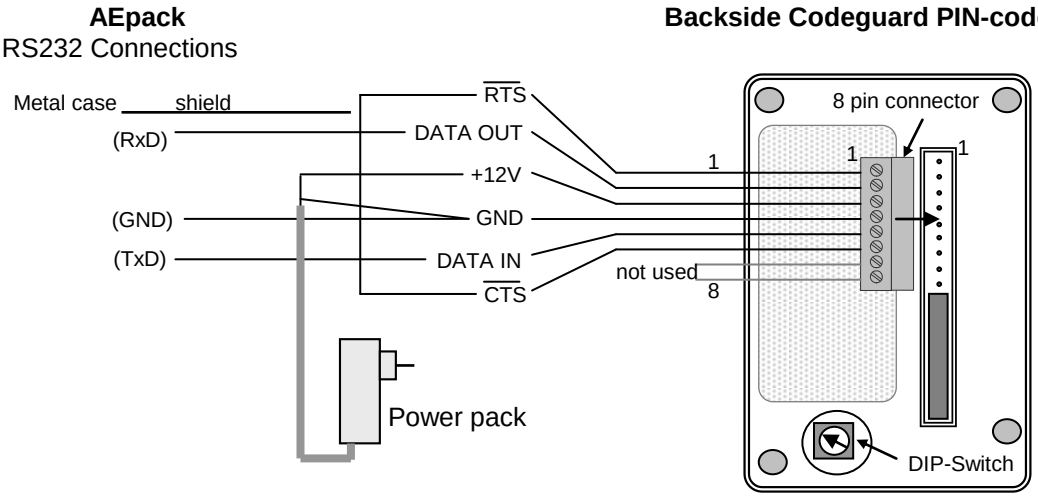
6.8.1. CodeGuard RS232 PIN-code terminal connections

For installation follow steps below:

Step 1: Install the mounting set on the correct position.
Attention: Cable length between Codeguard and AEpack is max. 15 meters (due to RS232).

Step 2: On Codeguard set DIP switch at backside to mode 'D' (1200Bd, Even parity)

Step 3: Make all connections as shown below, check also table below for availability of PIN code interface for these AEpacks.



Connections between Codeguard terminal (8 pin connector) and RS232 Interface on AEpack and Power pack:

8 pin connector Codeguard terminal	Function from Codeguard	Connect to
nc	Cable shield	Housing AEpack
1	*RTS	Codeguard pin 6
2	Tx	AEpack RxD
3	+12V	Power Pack +12V
4	GND	AEpack GND
		Power Pack GND
5	Rx	AEpack TxD
6	*CTS	Codeguard pin 1
7,8	not used	

Attention: Always check the hardware version and firmware of the AEpack:
Depending of the AEpack the RxD, GND and TXD can be on upper or lower connector.
If PIN code is possible is also depending on the **used hardware** (can be checked at the **PMS**) and the used firmware.

AEpack	PMS	Connections TxD, GND, RxD
AP1001	B of higher	Upper connector: 8,9,10
AP1002	B or higher	Upper connector: 8,9,10
AP1003	A or higher (check firmware)	Lower connector: 8,9,10
AP1005	No pincode possible	--
AP1007	A or higher (check firmware)	Upper connector: 8,9,10
AP1009	No pincode possible	--

Step 4: Check all connections before switching AEpack power ON and powering Codeguard terminal.

Step 5: After Codeguard is powered ON: display lights up with: *cc1 232 d* (representing the mode of the Codeguard) for some seconds. After this all keys light up with a '*8*' and Codeguard goes to the neutral situation in which only the led of '*' is ON.
If none of this information is seen or missing, check settings and/or connections.
If from neutral situation the '*' is pressed, a sound is generated and all keys light up with a '*d*'.

Step 6: Commision the AEpack using AEmon (see chapter 6.8.4)

Step 8: Check functioning of Codeguard PIN-code terminal by using an authorised badge with PIN-code and a non authorised badge using following scheme:

1. Neutral situation	• LED of '*' continuous red, door is closed
2. Present NON authorised badge	• No action on Codeguard, door remains locked
3. Present the authorised badge	• Codeguard lights up with keys randomly placed. Entering a key responds in a key-click sound and a short flash of the LED in the upper left. • After entering the correct PIN-code (ended with *#)* the Codeguard turns off and door is unlocked for the programmed unlock time • Entering a wrong PIN-code results in an error sound. You have to re-enter the PIN-code again The maximum number of attempts is 2. After the second time you have to represent your badge.

6.8.2. CodeGuard RS232 PIN-code terminal Product numbers

Product name	Nedap Prod. nr.
Codeguard PIN-code terminal	9829709
Surface mounting set	5460387
Flush mounting set	5460379
Power Pack for Godeguard	7655002

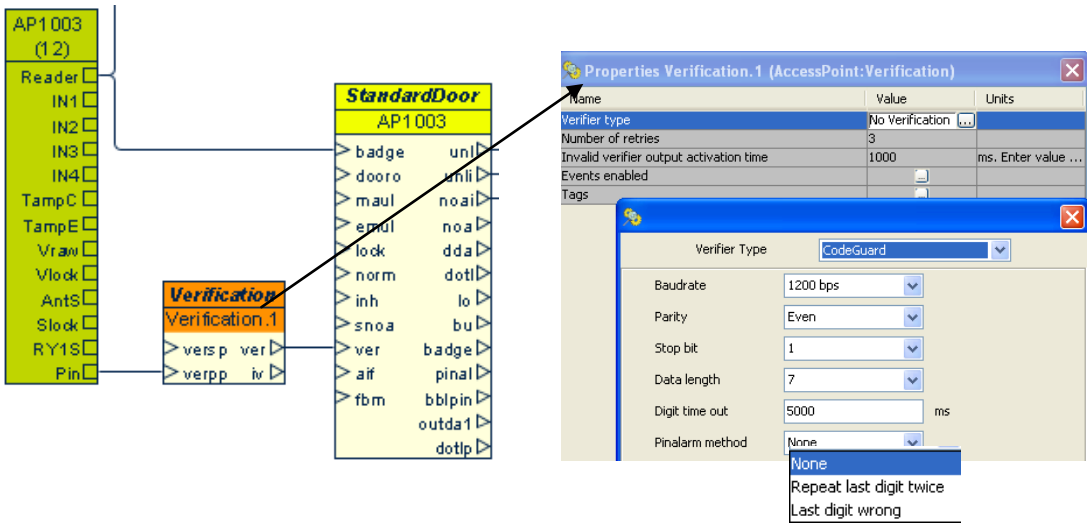
6.8.3. CodeGuard RS232 PIN-code terminal Earth connections

Take care that the earth connections are correct made. The cable must be shielded and the shield must make contact to the housing of the AEpack.

6.8.4. CodeGuard PIN-code terminal Commissioning

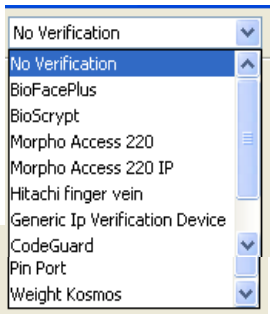
At AEmon the correct type of verification must be selected (see also chapter 6.3)

- *View – Configuration*, select the corresponding AEbc (type Access Point);
- Go to tab *Properties* and select *Verifier Type* : Code Guard, settings as below



- Digit Timeout: Time that CodeGuard digits are turned off after last number entered.
- Pinalarm method: How a silent alarm can be generated.

6.9. Other verification methods



Beside PIN other possibilities are (check Verification component in AEmon). These verification devices sometimes operates over a serial line or by IP.

With the *Generic IP Verification* a flexible solution is created to handle IP verification devices that are not explicitly been implemented in AEOS.
More information can be found in the AEbc descriptions

7. AP1003 / AP4x03 / AP6003 protocols

This chapter explains briefly all available software protocols which can be read by the AP1003 / AP4003 / AP6003 (check if the firmware is available for the used reader)
This protocol is the data format that the external readers supplies to the AP1003. No all protocols are available for AP1003, AP4003 and AP6003.

7.1. RS232 protocols

RS232 is split up into two different types: *RS232* and *RS232 Alpha*
This protocol is a simple protocol (printer format), without hand-shaking characters.
End of transmission is determined by one of the following:

- Timeout after 100msec no data is anymore received (there is no CR as termination is received)
- The transmitted data is **only** followed by **CR (**and LF) as terminating characters:
  CR : Carriage return : 0D hex
  LF : Line feed : 0A hex

7.1.1. RS232

Format: Label number: Range '0' ..'9' (030 hex – 039 hex), Max 20 bytes
9600 Baud No parity, 8 bits, 1 stop bit
Data is sent to AEpu as BCD
Use Generic Identifier type: Format BCD

7.1.2. RS232 Alpha nummeric

Format: Label number: Range 020 hex – 080 hex, Max 38 bytes
9600 Baud No parity, 8 bits, 1 stop bit
Data is sent to AEpu as ASCII
Use Generic Identifier type: Format Ascii-Alphanummeric

7.2. Omron protocols

Omron means behaviour as a magnetic card. The Omron code is according to standard ISO 7811/2, Track 2, with ABA format.

7.2.1. Omron data definition

The data information is included between start byte (start sentinal) and end byte (stop sentinal). After the stop sentinel a control byte (Long Redundancy Check) is transmitted to check the data.
A typical Omron code is built up as follows:

Byte Nr	Information	Nr of bytes	Data
1	Start sentinel	1	0B (hex)
….	data	x	Card data (decimal)
Last	Stop sentinel	1	0F(hex)
Last +1	LRC	1	Depending from above data

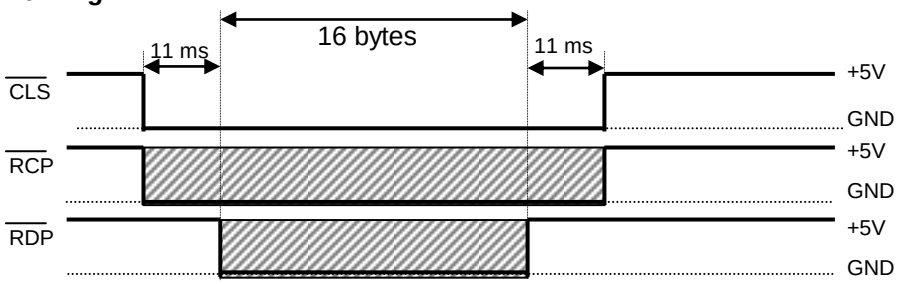
Each symbol (byte) consists out of 5 bits including a parity bit (ODD). First the LSB is sent, finally the parity bit is sent.
LRC : Vertical parity (EVEN) over the 4 data bits over all bytes including start and stop sentinel (excluding the parity bit)

The complete message consists of the following:
Bit position : `12345 67890 12345 67890 ….12345 67890 12345 67890`
Bit contents : `BBBBP DDDDP DDDDP DDDDP ….DDDDP DDDDP EEEEP LLLLP`

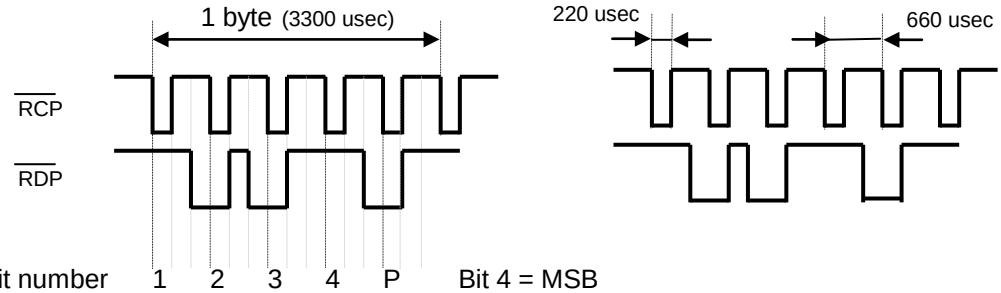
Legend : B : Begin Sentinel Character (4 bits: fixed B hex)
P : Parity ODD over preceding 4 bits
D : Data (x bytes)
E : End Sentinel Character (4 bits: fixed F hex)
L : Longitudinal Redundancy Check Character

Format: Depends on used reader
AEpack data: 0202 00 0987023456
0202: Omron type ISO Track 2
00: If first byte is 1, second byte is part of data,
If first byte is 0, second byte in not used
Normally these bytes are 00 and not included into the data
0987023456 Data read, with AEmon (AccessPoint) can be split up into customer code (optional) and card number

7.2.2. Omron timing



Below for one byte the bit timing in specified: Each bit consists out of one period low (220usec) and two periods high (440usec). The bit times have a resolution of 10%.
Data valid on falling edge of /RCP. (CLS signal is not used.) For reading data timing is not critical.



Bit number	1	2	3	4	P	Bit 4 = MSB
Bit data	0	1	1	0	1	Data = 06 Hex

Example Timing: Clock period : 660 usec
Clock high : 440 usec
Clock low : 220 usec
Resolution : 10%

7.2.3. Omron 200 max

Identical to standard Omron protocol. Useful if reader uses e.g. a hexadecimal dataformat or if format is not known.
Shows amount of data bits read, message includes amount of data bits. Differences to standard:

- Data must start with Pre-amble (16 bits '1') and end with Post-amble (16 bits '1')
- No check on
  - -Begin Sentinel Character (4 bits: fixed B hex)
  - -End Sentinel Character (4 bits: fixed F hex)
  - -Longitudinal Redundancy Check Character
- Parity is checked
- Data included amount of data bits is sent using the Omron protocol

Format: Depends on used reader
AEpack data: 0204 38 B1F50C991B4DF4
0204: Omron type 200 max
38 Amount of bits (hex) → 56 bits
B1F50C991B4DF4 Data read (incl. start sentinel B, end sentinel F and LRC (4))

Data is one string, using Generic Identifier, this format can be split up into Customercode and labelnumber if desired. Or use Customercode to check on the amount of bits received.

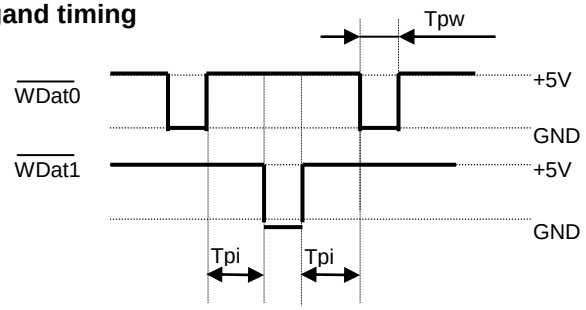
7.3. Wiegand protocols

Several Wiegand formats can be used, check the appropriate firmware:

- Wiegand 26
- Wiegand 32
- Wiegand 32bin / Wiegand 32binRev
- Wiegand 32C1000 / Wiegand 26C1000
- Wiegand 37 / Wiegand 37KPM5p2
- Wiegand 64bin / Wiegand 128bin
- Wiegand 128

Wiegand data is been received at RDP and RCP, signals are active low.

7.3.1. Wiegand timing



Timing:		Min	Max
Tpw	Pulse Width Time	20 usec	100 usec
Tpi	Pulse Interval Time	200 usec	20 msec

7.3.2. Wiegand 26

Format: Facility code: 8 bits: range: 0..255
Label number: 16 bits: range: 1..65535
Parity: 2 bits
AEpack data: 0301 0A0039 (cust. code 0A, cardnr: 57)

7.3.2.1. Wiegand 26 data definition

The Wiegand 26-bit format consists of a parity bit, followed by eight facility code bits, followed by 16 label number code bits and a final parity bit. The first parity bit shall create even parity when combined with the next twelve bits. The last parity bit shall create odd parity when combined with the remaining 12 bits.
The data is represented in bits. (unsigned binary coded)
The complete message consists of the following:

```
Bit position  :  1 23456789 0123456789012345 6
Bit contents  :  P FFFFFFFF NNNNNNNNNNNNNNNN P

Parity (even):  P FFFFFFFF NN
       (odd) :                 NNNNNNNNNNNNNNN P
```

Legend : F : Facility code (8 bits: range: 0..255)
N : Label number (16 bits: range: 1..65535)
P : Even parity over the first 12 bits
Odd parity over the next 12 bits

7.3.3. Wiegand 26C1000

Combined format for Wiegand 26 and Wiegand C1000. Data received is sent in corresponding format.
Format: Wiegand 26 (see 7.3.2) or Wiegand C1000 (see 7.3.7)

7.3.4. Wiegand 32

Format: Facility code: 16 bits: range: 0..65535
Label number: 16 bits: range: 1..65535

7.3.4.1. Wiegand 32 data definition

The Wiegand 32-bit format consists 16 facility code bits, followed by 16 label number code bits. There are no parity bits.
The data is represented in bits. (unsigned binary coded)

The complete message consists of the following:

```
Bit position  :  1234567890123456 7890123456789012
Bit contents  :  FFFFFFFFFFFFFFFF NNNNNNNNNNNNNNNN
```

Legend : F : Facility code (16 bits: range: 0-65535)
N : Label number (16 bits: range: 1..65535)
AEpack data: 0302 00FF 01E23 (cust. code 255, cardnr: 7715)

7.3.5. Wiegand 32bin

Format: Label number: 32 bits: range: 1..FF FF FF FF
Data is represented in hex, first bit first.
Data read is e.g. *E2 9C B8 1E* is presented as card number *E2 9C B8 1E*
AEpack data: 0304 E29CB81E

Data is one string, using Generic Identifier, this format can be split up into Customercode and labelnumber if desired.

7.3.6. Wiegand 32binRev

Format: Label number: 32 bits: range: 1..FF FF FF FF
Data is represented in hex, first bit last.
Data read is e.g. *1E B8 9C E2* is presented as card number *E2 9C B8 1E*
AEpack data: 0304 1eb89ce2

Data is one string, using Generic Identifier, this format can be split up into Customercode and labelnumber if desired.

7.3.7. Wiegand 32C1000

Data with HID Corporate 1000 (C1000) protocol will be sent using a Wiegand 32 protocol. With AEOS Generic type Facility Code (Customer code) and Card number can be filtered.

Card: HID (Corporate 1000 protocol)
Data format: P 1 bit Even Parity over all E-bits (22 bits)
P 1 bit: Odd Parity over all O-bits (22 bits)
F 12 bits Facility code hex (0..4096)
N 20 bits Card number hex (0..1048575)
P 1 bit: Odd Parity over all O-bits (34 bits)

Parity is determined over several bits.

Format: Label number: 32 bits: range: 1..FF FF FF FF
Data is represented in hex, first bit last.
Data read e.g by AEpack: 0309 1E B8 9C E2 *1E B8 9C E2* must be split up using the Generic Identifier in:

- 12 bits Facility code 1EB Customer code: (1EB)
- 20 bits Card number 8 9C E2 Card number (564450)

AEpack data: 0309 1E B8 9C E2

In AEmon use Generic Identifier for retrieving card nr and customer code:

Badge number			Customer code		
Format	Hexadecimal	Numeric	*Format*	Hexadecimal	Hex
Bit position	12		*Bit position*	0	
Length	20		*Length*	12	

At AEserver fill in as IdentifierType: Generic, settings see AEmon.

7.3.8. Wiegand 36

Format: Wiegand36

Bit contents : `P FFFFFFFF FFFFFFFF NNNNNNNN NNNNNNNN ZZ P`

P	1 bit	Odd Parity over bits 1 – 18 (17 bits)
F	16 bits	Facility code hex (0..65535)
N	16 bits	Card number hex (0..65535)
Z	2 bits	(unknown, Zero)
P	1 bit	Even Parity over bits 19– 35 (17 bits)

Wiegand 36 can be read with universal software *APx003wieg128max*

AEpack data:
```
0308 24 001900b218      cardnr: 05699 (customer code 800)
0308 24 001900b221      cardnr: 05700 (customer code 800)
```

In AEmon use Generic Identifier for retrieving card nr and customer code:

Badge number			Customer code		
Format	Hexadecimal	Numeric	*Format*	Hexadecimal	Numeric
Bit position	29		*Bit position*	13	
Length	16		*Length*	16	

At AEserver fill in as IdentifierType: Generic, settings see AEmon, as Customer code: 800 (numeric)

7.3.9. Wiegand 37

Format: Label number: 35 bits: range: 1..34359738368

7.3.9.1. Wiegand 37 data definition

The Wiegand 37-bit format consists of one number, no facility code. There are 2 parity bits.
The data is represented in bits. (unsigned binary coded)

The complete message consists of the following:

```
Bit position  :  1 23456789 0123456789 0123456789 0123456 7
Bit contents  :  P NNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNN P
Parity (even) :  P 12345678 9012345678
     (odd)    :                      1 2345678901 2345678 P
```

Legend : N : Label number (35 bits)
P : Even parity over the first 18 bits
Odd parity over the next 18 bits (starts from bit 18)

AEpack data: 0303 0539C13B2A

7.3.10. Wiegand 37KPM5p1

Format: Label number: 35 bits: range: 1..34359738368
Replaces Wiegand37KPM5

7.3.10.1. Wiegand 37KPM5p1 data definition

The Wiegand 37-bit format consists of one number, no facility code. There are 2 parity bits.
The data is represented in bits. (unsigned binary coded)

The complete message consists of the following:

```
Bit position  :  1 23456789 0123456789 0123456789 0123456 7
Bit contents  :  P NNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNN P
```

```
Parity (even) : P 12345678 9012345678
        (odd) :                    1 2345678901 2345678 P
```

Legend : N : Label number (35 bits)
P : Even parity over the first 18 bits
Odd parity over the next 18 bits (starts from bit 18)
Remark: Replaced by *Wiegand37_543v203*

7.3.11. Wiegand 37KPM5p2

Format: Label number: 35 bits: range: 1..34359738368
Replaces Wiegand37PRM5 (but includes now PINcode)

7.3.11.1. Wiegand 37KPM5p2 data definition

The Wiegand 37-bit format consists of one number, no facility code. There are 2 parity bits.
The data is represented in bits. (unsigned binary coded)
The complete message consists of the following:

```
Bit position  : 1 23456789 0123456789 0123456789 0123456 7
Bit contents  : P NNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNN P
Parity (even) : P 12345678 9012345678
        (odd) :                    1 2345678901 2345678 P
```

Legend : N : Label number (35 bits)
P : Even parity over all 35 bits
Odd parity over all 35 bits

7.3.12. Differences between Wiegand 37KPM5p1 and p2

Wiegand37:	Up from *Wiegand37_543v203* PIN-code possible No PIN code possible, parity see above
Wiegand37KPM5p1:	PIN code possible (PIN code keys: inversed nibbles), parity see above Was before (2006) *Wiegand37KPM5* Replaced by *Wiegand37_543v203*
Wiegand37KPM5p2:	Up from *Wiegand37KPM5p2 _543v2.03* PIN-code possible PIN code possible (PIN code keys: inversed nibbles), parity see above Was before (2006) Wiegand37PRM5, but now with PIN code)

7.3.13. Wiegand 64bin

Format: Label number: 64 bits: range: 1.. FF FF FF FF FF FF FF FF
Data is represented in hex, first bit first.
Data read is e.g. *E2 9C B8 1E* is presented as card number *E2 9C B8 1E*

Data is one string, using Generic Identifier, this format can be split up into Customercode and labelnumber if desired.

AEpack data: 0307 0000012AA8280A0B

7.3.14. Wiegand 128bin

Format: Label number: 128 bits: range: 1.. FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
Data is represented in hex, first bit first.
Data read is e.g. *66 03 30 55 50 AB BA 0D EA D0 FE ED 01 23 45*

Data is one string, using Generic Identifier, this format can be split up into Customercode and labelnumber if desired.

AEpack data: 0308 *6603305550ABBA0DEAD0FEED012345*

7.3.15. Wiegand 128max (named Wiegand128 till version 2.02)

Format: Label number: Up to 128 bits:
range: 1.. FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
Data is represented in hex, starting with 2 bytes for the amount of bits, first bit first.
Data read is e.g. *019B7BA8*
Data is one string, using Generic Identifier, this format can be split up into Customercode and labelnumber if desired.

AEpack data: 0308 *1A 019B7BA8*
1A : 26 bits (1A = 26)
019B7BA8 : Data = 01100110110111101110101000 binair
This protocol is used for determining which communication protocol can be used for unknown readers.

Up from version v2.02 (AP1003wieg128_543v202) the Wiegand 128 protocol supports PIN-code. If the received data is a valid PIN-code format (see chapter 6) the PIN-code is accepted. If not the data is shown using above protocol.
Versions before 2.02: If a PIN-code is received, this is translated to the string data above (so it can be used for determining how the PIN-code is sent).

7.3.16. Wiegand 128maxMultiPin

Identical to Wiegand 128max (see chapter 7.3.15), only the PIN code is handled different:
Data of 4, 8, 12, 16, 20, 24, 28 and 32 bits data is processed as PIN-code.

AEpack data: *0308 34 00004417805415* cardnumber
AEpack PIN data *31323334* pincode (4 digits)
From reader received 0308 10 1234 (converted to 31323334)

8. AP1003 / AP4x03 firmware

Below the standard firmware is given: (up from 4-2007)

	AP1003	AP4003	Remark
Omron	…Omron	…Omron	
RS232	…RS232 (..ALPHA)	…RS2323 (..ALPHA)	Serial interface
RS232 PIPS	…RS232PIPS		PIPS number plate recognition
RS232 IButton	…RS232IButton	…RS2323IButton	IButton interface
Wiegand 26	…wieg26	…wieg26	
Wiegand 32	…wieg32	…wieg32	16 bits fact code + 16 bits cardnr
Wiegand 32 bin	…wieg32bin	…wieg32bin	Mifare CSN (Philips)
Wiegand 32 binRev	…wieg32binRev	…wieg32binRev	Mifare CSN Reversed (HID)
Wiegand C1000	…wieg32C1000	…wieg32C1000	HID C1000 protocol
Wiegand 37	…wieg37	…wieg37	
Wiegand 37Deister PRM5	...wieg37KPM5p2 …wieg37deisterPRM5	...wieg37KPM5p2	Deister reader
Wiegand 37Deister KPM5	…wieg37 …wieg37deisterKPM5 …wieg37KPM5p1	…wieg37	Deister reader with PIN pads
Wiegand 64 bin	…wieg64bin	…wieg64bin	64 bits hex data
Wiegand 128 bin	…wieg128bin	…wieg128bin	128 bits hex data
Wiegand 128 max	…wieg128max	…wieg128max	All data received, incl amount
XS	…XS		MD300 XS integration

The availability of the firmware can be checked on www.nedap.net, AEOS CD or AEmon. If the firmware is not available contact your Nedap representative.

9. Readers tested with AP1003 / AP4x03 / AP6003

In this chapter tested readers are shown, included their settings. Some readers are tested for specific projects.

9.1. General information

9.1.1. Remarks

- Readers are tested at the then available version of AEOS. Newer versions can have other possibilities.
  In case of problems always check the logfile (see also chapter 2.7)
- Using the Generic Identifier offers much more flexibility than some standard protocols. (see also chapter 2.8). Keep in mind that sorting out how the Generic Identifier must be programmed takes some more expertise.
- Up from 4-2007 (with AEOS 2.1) most protocols are now available including PIN-code and for both the AP1003 and AP4003.
- In most situations AP1003, AP4003 and AP6003 can be used, please check the availability of the firmware to be used.
- Of course many other readers can be connected, just check the used protocol!

9.1.2. Different formats

For most situations the data can be read using a specific interface (e.g. Wiegand32), beside that also the Generic Identifier type (see chapter 2.8) can be used. With the Generic Identifier a mix of different readers can be applied.

9.1.3. AP1003 / AP4x03 / AP6003 Check identifier settings using AEpu logfile

Following steps must be valid to check a card in the AEpu logfile:

- Card must be recognized at the used reader (LED on reader blinks?)
- Card data must be transferred to the AEpack (LED's on AEpack at data signals blinks?)
- ID led blinks:Data format is valid for AEpu
- Relay is activated: Card number is valid

If not, check logfile and eventdata at AEmon to retrieve more information.

9.1.4. Description of used fields at reader information part:

9.1.4.1. Reader / Interface / carddata

Reader : Deister KPRM5 This is the kind of reader that is connected to AEOS
Reader firmware : If know, the firmware and version of the third party reader
I*nterface : Wiegand37* How the third party reader interfaces with the AEpack
Card : Mifare Kind of card used
Data format : Customercode, Badge number How the data is stored on the card

9.1.4.2. Software, Firmware and Settings

AEOS: Version 2.0.5 AEOS version which is used to test this special
AEpack: AP1003Wieg37_543v201 AEpack firmware for AP1003 and AP4003
AP4003Wieg37_543v202
AEpack data: 0107 01 29100 8070731 What data is been sent from the AEpack (AP1003/4003) to the AEpu

***AEmon**: IdentifierType: Generic* AEmon settings for Identifier, Verification, etc
***AEserver**: IdentifierType: Generic* Identifier settings at AEmon

9.2. Deister PRX5, KPM 5/2, PRM5/2 Wiegand interface

Deister readers reading Mifare cards, optional including PINcode terminal connected using Wiegand interface. Program in Deister reader detemine functionality.
Remark: Deister readers can cause interference on the data lines, specially with short cabling. This can be improved by adding a 1nF condensator between the Data lines (D0 / D1) and GND

9.2.1. Reader / Interface / Carddata

Reader: Deister: PRX5, KPM5/2 PRM5/2. Program 741
Interface: Wiegand37
Carddata: Mifare ISO 14443A CSN Mifare CSN number.

9.2.2. Firmware and Settings AEOS 2.0.4

APx003: APx003wieg37_543v203 Including PIN
APx003: APx003wieg37KPM5p2_54v203 Including PIN
AP1003: AP1003Wieg37 No PIN possible
AP4x03: AP4003Wieg37 No PIN possible
AP1003: AP1003wieg37deisterKPM5 Including PIN
AP1003: AP1003wieg37KPM5p1 / p2 Including PIN

9.2.2.1. Identifier settings using Wiegand

AEmon: IdentifierType: Wiegand
Verifier Type: PinPort
Reader data: 030300E645C1B1
Cardnumber is E645C1B1 hex, must be entered as Numeric (3863331249)
AEserver: IdentifierType: Wiegand37 Numeric (data entered as Numeric at AEOS front-end)
Cardnumber at frontend: Numeric (3863331249)

9.2.2.2. Identifier settings using Generic Identifier

AEmon: IdentifierType: Generic, see below
AEserver: IdentifierType: Generic, take settings from AEmon

Badge number			**Customer code (none)**	
Format	Hexadecimal	*Hex*		
Bit position	8			
Length	32 bits			

Cardnumber at frontend: Hexadecimal (E645C1B1)

9.2.3. Connections

Signal	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
1 to pin 6	Blue				Interconnection on reader
2 GND	Black		14	14	
3 +8..14VDC	Red	12		External	
4 D1	White		11	11	RDP
5 D0	Green		12	12	RCP
6 to pin 1	Blue				Interconnection on reader
7 Green led	Yellow	14		13	UL*
8 Red led	Purple	15		15	NA*

9.3. HID EPIC 315 Wiegand interface

9.3.1. Reader / Interface / Carddata

Reader: HID EPIC 315
Interface: Wiegand
Carddata: Wiegand26 (depending of type card).

9.3.2. Firmware and Settings AEOS 2.0.4

AP1003: AP1003Wieg26 v1.xx
AP1003Wieg26 v2.05
AP4x03: AP4003Wieg26 v2.xx

9.3.2.1. Identifier settings using Wiegand

AEmon: IdentifierType: Wiegand
Reader data: 0301 FF F6 DE
Customercode: FF = 255
Cardnumber: F6DE = 63198
AEserver: IdentifierType: Wiegand26 Numeric

Printable		*Print receipt*	
Identifier length			
Identification Type	Numeric		
Identifier min. value	1	*Identifier max. value*	65535
Has different intern.		*Add leading zeroes*	
Series	Wiegand26		
Customer code	255		

Cardnumber and customercode at frontend: Numeric (63198 / 255)

9.3.2.2. Identifier settings using Generic Identifier

AEmon: IdentifierType: Generic, see below
AEserver: IdentifierType: Generic, take settings from AEmon

Badge number			**Customer code (none)**	
Format	Hexadecimal	*Hex*		
Bit position	8 bits			
Length	16 bits			

Cardnumber at frontend: Hexadecimal (F6DE)

9.3.3. Connections

Signal		**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
			Upper	**Lower**		
1	GND	Black	9		9	
2	+5 V	Red	11		1	
3	RDP	White		11	11	
4	RCP	Green		12	12	

9.4. HID IQcard Mifare reader Wiegand interface

9.4.1. Reader / Interface / Carddata

Reader: HID IQcard Mifare reader 6055 ABN00
Interface: Wiegand32
Carddata: Mifare ISO 14443A CSN Mifare CSN number

9.4.2. Firmware and Settings AEOS 2.0.4

AP1003: AP1003Wieg32bin v1.xx
AP1003Wieg32bin v2.xx
AP4x03: AP4003Wieg32bin v2.xx

AEmon: IdentifierType: Wiegand
Reader data: 0304 E645C1B1
Cardnumber: E645C1B1
AEserver: IdentifierType: Wiegand32 binary

Printable		*Print receipt*	
Identifier length			
Identification Type	Hexadecimal		
Identifier min. value	1	*Identifier max. value*	FFFF FFFF
Has different intern.		*Add leading zeroes*	
Series	Wiegand32 binary		
Customer code			

Cardnumber at frontend: Hexadecimal (E645C1B1)

9.4.3. Connections

Signal		Colour	AP1003 connector		AP4x03 connector	Remarks
			Upper	Lower		
1	GND	Black	9		9	
2	+5 V	Red	11		1	
3	RDP	White		11	11	
4	RCP	Green		12	12	

9.5. HID ProxPoint Plus 6005B Wiegand interface

Settings identical to chapter **9.7**; check card format for correct firmware!

9.6. HID HID ProPro 5355AGN Pin Wiegand interface

Settings identical to chapter **9.7**; check card format for correct firmware!

9.7. HID MiniProx readers

Remark: The format on the used cards determines which firmware must be used.

9.7.1. Reader / Interface / Carddata

Reader: HID MiniProx 5365 CGT00
Interface: Wiegand
Carddata: HID RF prox Wiegand 37 / Wiegand26

9.7.2. Firmware and Settings AEOS 2.0.4 Wiegand 37

Carddata: HID RF prox Wiegand 37
AP1003: AP1003Wieg37 v1.xx
AP1003Wieg37 v2.xx
AP4x03: AP4003Wieg37 v2.xx

9.7.2.1. Identifier settings using Wiegand

AEmon: IdentifierType: Wiegand
Reader data: 0303 0000 46 66 33
Cardnumber: 466633 = 4613683
AEserver: IdentifierType: Wiegand37 Numeric (data entered decimal at AEOS front-end)

Printable		*Print receipt*	
Identifier length			
Identification Type	Numeric		
Identifier min. value		*Identifier max. value*	
Has different intern.		*Add leading zeroes*	
Series	Wiegand37		
Customer code			

Cardnumber at frontend: Numeric (4613683)

9.7.2.2. Identifier settings using Generic Identifier

AEmon: IdentifierType: Generic, see below
AEserver: IdentifierType: Generic, take settings from AEmon

Badge number			**Customer code**		
Format	Hexadecimal	*Hex*	*Format*	Hexadecimal	*Hex*
Bit position	16 bits		*Bit position*	0	
Length	24 bits		*Length*	16 bits	

Cardnumber and customercode at frontend: Hexadecimal (466633 / 0)

9.7.3. Firmware and Settings AEOS 2.0.4 Wiegand 26

AP1003: AP1003Wieg26 v1.xx
AP1003Wieg26 v2.xx
AP4x03: AP4003Wieg26 v2.xx

AEmon: IdentifierType: Wiegand
Reader data: 0301 0A 0036
Customercode: 0A = 10
Cardnumber: 0036 = 54
AEserver: IdentifierType: Wiegand 26 Numeric (data entered decimal at AEOS front-end)

Printable		*Print receipt*	
Identifier length			
Identification Type	Numeric		
Identifier min. value	1	*Identifier max. value*	65535
Has different intern.		*Add leading zeroes*	
Series	Wiegand26		
Customer code	10		

Cardnumber and customercode at frontend: Numeric (54 / 10)

9.7.4. Connections

Signal		Colour	AP1003 connector		AP4x03 connector	Remarks
			Upper	Lower		
1	GND	Black	9		9	
2	+5 V	Red	11		1	
3	RDP	White		11	11	
4	RCP	Green		12	12	

9.8. HID IClass RK40 PINPad

Remark: The format on the used cards determines which firmware must be used (Wiegand or Mifare).

9.8.1. Reader / Interface / Carddata

Reader: HID IClass 6130 AKN
Interface: Wiegand26
Carddata: HID RF prox Wiegand 26 / Mifare CSN

9.8.2. Firmware and Settings AEOS 2.0.4 Wiegand 26

AP1003: AP1003Wieg26 v1.xx Including PIN
AP1003Wieg26 v2.xx Including PIN
AP4x03: AP4003Wieg26 v2.xx Including PIN

AEmon: IdentifierType: Wiegand
Verification: PinPort
Reader data: 0301 0A0001
Customercode: 0A = 10
Cardnumber: 0001 = 1 (data entered decimal at AEOS front-end)
AEserver: IdentifierType: Wiegand 26 numeric

Printable		*Print receipt*	
Identifier length			
Identification Type	Numeric		
Identifier min. value	1	*Identifier max. value*	65535
Has different intern.		*Add leading zeroes*	
Series	Wiegand26		
Customer code	10		

Cardnumber at frontend: Numeric (1)

9.8.3. Firmware and Settings AEOS 2.0.4 Mifare CSN

AP1003: AP1003Wieg32binRev v1.xx Including PIN
AP1003Wieg32binRev v2.xx Including PIN
AP4x03: AP4003Wieg32binRev v2.xx Including PIN

AEmon: IdentifierType: Wiegand
Verification: PinPort
Reader data: 0304 E645C1B1
Cardnumber: E645C1B1 (data entered hexadecimal at AEOS front-end)
AEserver: IdentifierType: Wiegand32 binary

Printable		*Print receipt*	
Identifier length			
Identification Type	Hexadecimal		
Identifier min. value	1	*Identifier max. value*	FFFF FFFF
Has different intern.		*Add leading zeroes*	
Series	Wiegand32 binary		
Customer code			

Cardnumber at frontend: Hexadecimal (E645C1B1)

9.8.4. Connections

Signal		Colour	AP1003 connector		AP4x03 connector	Remarks
			Upper	Lower		
1	GND	Black	9		9	
2	+5..16V	Red	12		1	12V on AP1003, 5V on AP4x03
3	RDP	White		11	11	
4	RCP	Green		12	12	

9.9. Legic PR74 RWC (LEG-SC-RW-002), Triple Eye

Reader not tested, is not available anymore at Groenlo.

9.9.1. Reader / Interface / Carddata

Reader: Legic PR74 RWC (LEG-SC-RW-002)
Interface: Omron
Carddata: ….

9.9.2. Firmware and Settings AEOS 2.x

AP1003:
AP4x03:

AEmon: IdentifierType: Wiegand
AEserver: IdentifierType: Wiegand

Printable		*Print receipt*	
Identifier length			
Identification Type			
Identifier min. value		*Identifier max. value*	
Has different intern.		*Add leading zeroes*	
Series	Wiegand??		
Customer code			

9.9.3. Connections

Signal	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
2 Red led	Purple	15		15	NA*
3 Green led	Yellow	14		13	UL*
4 RCPI	Green		12	12	RCP
5 RDPI	White		11	11	RDP
6 CLSI	Blue		16	--	CLS not available on AP4x03
9 GND	Black		14	14	
10 +12VDC	Red	12		External	

9.10. Omron 3S4YR-HNR-CMH-PCB Omron interface

Magnetic card reader. RCP, RDP and CLS

9.10.1. Reader / Interface / Carddata

Reader: Omron 3S4YR-HNR-CMH-PCB
Interface: Omron
Carddata: Magnetic stripe track 2. Cardnumber can be anywhere in the received string. Can be retrieved by changing the AEmon settings.

9.10.2. Firmware and Settings AEOS 2.0.4

AP1003: AP1003omron_v107
AP1003omron_v204
AP4x03: AP4003omron_v202

AEmon: IdentifierType: Omron
Reader data: 0202 00 023456
Cardnumber: 023456 (00 is skipped as part of the card number)
AEserver: IdentifierType: Omron

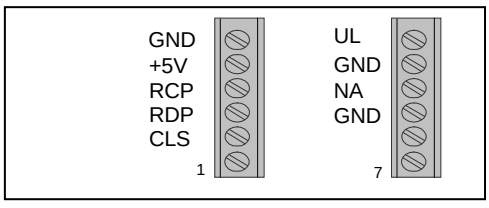
Printable		*Print receipt*	
Identifier length	6 *		
Identification Type	Omron		
Identifier min. value	1	*Identifier max. value*	999999 *
Has different intern.		*Add leading zeroes*	Yes *
Customer code			

*****: Omron is interpreted as a string; identifier length is depending of the card data and the settings at AEmon. Leading zero's are only for the user's convenience: so he has not to enter all decimals. Customer code is only if this is also stated at AEmon.

9.10.3. Connections

Signal	Colour	AP1003 connector		AP4x03 connector	Remarks
		Upper	Lower		
6 GND	Grey	9		9	
5 5 V	Rose	11		1	
3 RDP	Brown		11	11	
4 RCP	White		12	12	
12 Led Grn +	Green		13	-	Led on AP4x03 uses common
11 Leds -	Black		14	-	anode
10 Led Red +	Yellow		15	-	
2 CLS	Blue		16	-	AP4x03 uses no CLS
9 & 11					Interconnection

Metal housing: (check also documentation)



9.11. Opticon NSR120 Barcode interface

Barcode reader. Not tested at Groenlo

9.11.1. Reader / Interface / Carddata

Reader: Opticon NSR120 (slot reader)
Interface: RS232
Carddata: Barcode

9.11.2. Firmware and Settings AEOS 2.x

AP1003: AP1003rs232 v1.xx
AP1003rs232 v2.xx
AP4x03: AP4003rs232 v2.xx

AEmon: IdentifierType:
AEserver: IdentifierType:

Printable		*Print receipt*	
Identifier length			
Identification Type			
Identifier min. value		*Identifier max. value*	
Has different intern.		*Add leading zeroes*	
Series			
Customer code			

9.11.3. Connections

Signal		**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
			Upper	**Lower**		
7	GND	Black	9		9	
3	Tx	Green	10		10	Rx on AEpack
25	5 V	Red	11		1	

9.12. INDALA FlexPass FPC-0500A+

Remark: Below the settings for Wiegand 26 are given. Depending of the used software in the Indala reader the dataformat can be different (e.g. Wiegand 27)

9.12.1. Reader / Interface / Carddata

Reader: Indala FlexPass FPC-0500A+
Interface: Wiegand 26
Carddata: Wiegand 26

9.12.2. Firmware and Settings AEOS 2.0.4

AP1003: AP1003Wieg26 v1.xx
AP1003Wieg26 v2.xx
AP4x03: AP4003Wieg26 v2.xx

AEmon: IdentifierType: Wiegand
Reader data: 0301 FF0043
Customercode: FF = 255
Cardnumber: 0043 = 67 (data entered decimal at AEOS front-end)
AEserver: IdentifierType: Wiegand

Printable		*Print receipt*	
Identifier length			
Identification Type	Numeric		
Identifier min. value	1	*Identifier max. value*	65535
Has different intern.		*Add leading zeroes*	
Series	Wiegand26		
Customer code	255		

Cardnumber and customercode at frontend: Numeric (67 / 255)

9.12.3. Connections

Signal	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
Red led	Orange	15		15	NA*
Green led	Brown	14		13	UL*
Clock	White		12	12	RCP
Data	Green		11	11	RDP
GND	Black		14	14	
+5VDC	Red	12		1	5V DC

9.13. KeyProcessor SL-4 Magnetic OMRON T2

Magnetic card reader.

9.13.1. Reader / Interface / Carddata

Reader: Key processor SL-4
Interface: Omron
Carddata: Magnetic stripe track 2

9.13.2. Firmware and Settings AEOS 2.0.4

AP1003: AP1003Omron v1.xx
AP1003Omron v2.xx
AP4x03: AP4003Omron v2.xx

AEmon: IdentifierType: Omron (0,8,0,0) (8: In this case labelnumer is 8 long)
Reader data: 0202 00 29748446
00 is no part of badgenumber
Cardnumber: 29748446 (data entered decimal at AEOS front-end)
AEserver: IdentifierType: Omron

Printable		*Print receipt*	
Identifier length	8		
Identification Type	Numeric		
Identifier min. value	1	*Identifier max. value*	99999999
Has different intern.		*Add leading zeroes*	
Series	Omron		
Customer code			

Cardnumber at frontend: Numeric (29748446)
Remark: Cardnumber can be split up if necessary into Custumer code and cardnumber.
For doing this the settings at AEmon must also be changed.

9.13.3. Connections

Signal	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
GND	Brown	9		9	+ 2* white
5 V	Rose	11		1	
RDP	Green		11	11	
RCP	Yellow		12	12	
CLS	Blue		16	-	AP4x03 uses no CLS

9.14. DALLAS IButton DS1971 + converter DS9097

Dallas iButton reader. Uses converter (DS9097U-S09) to convert the 1-wire communication of Dallas reader to RS232.

9.14.1. Reader / Interface / Carddata

Reader: Dallas iButton
Interface: RS232
Carddata: iButton fixed number

9.14.2. Firmware and Settings AEOS 2.0.4

AP1003: AP1003RS232IButton v2.xx
AP4x03: AP4003RS232IButton v2.xx

AEmon: IdentifierType: Generic, see below
Reader data: 0109 C600000012145B814
AEserver: IdentifierType: Generic, take settings from AEmon

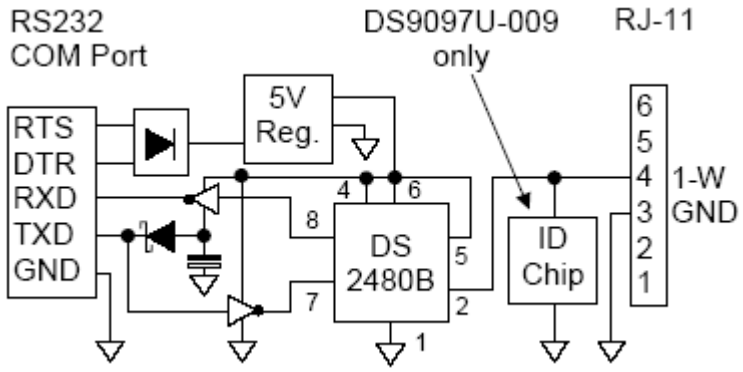
Badge number			**Customer code (none)**	
Format	Hexadecimal	*Numeric*		
Bit position	8			
Length	48			

Remark: Above settings gives only a part of the IButton data.
Cardnumber at frontend: Numeric (29748446)

9.14.3. Connections

Signal on DIN 7pin	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
2 RxD			8	8	
3 TxD			10	10	
5 GND			9	9	
4 DTR			11	1	DTR and RTS are used for powering
7 RTS			11	1	the converter (+ 5V DC)

Connections on Converter:
(Comm. settings: 9600 N 8 1)



9.15. Nedap Transit Reader

Nedap Transit readers can be connected on different ways to the AEOS readers:

- Using HF output of Transit and connect this to antenna input of AP1001, AP1002, AP1009 or AP4x01
  The Transit reader is now seen as an antenna for these AEpacks. Only the Nedap data is been transferred, combi boosters cannot be read on this way.
  Badge data is default Nedap XS (not explained below).
- Use Wiegand on Transit (P81 firmware), connect to AP1003, AP4x03.
  Using Wiegand gives you the opportunity to read also the combi-booster information.
  More detailed information can be found below.

9.15.1. Reader / Interface / Carddata

Reader: Transit PS270 (P81 firmware)
Interface: Wiegand26 or Wiegand37 (DIP switch setting on PS270)
Carddata: Depending of used label at Transit

9.15.2. Firmware and Settings AEOS 2.0.4

AP1003: AP1003Wiegand26 v1.xx AP1003Wiegand37 v1.xx
AP1003Wiegand26 v2.xx AP1003Wiegand37 v2.xx
AP4x03: AP4003Wiegand26 v2.xx AP1003Wiegand37 v2.xx

AEmon: IdentifierType: Wiegand
Reader data: 0301 FF0043
Customercode: FF = 255
Cardnumber: 0043 = 67 (data entered decimal at AEOS front-end)
AEserver: IdentifierType: Wiegand

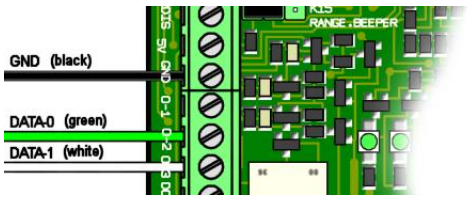
Printable		*Print receipt*	
Identifier length			
Identification Type	Numeric		
Identifier min. value	1	*Identifier max. value*	65535
Has different intern.		*Add leading zeroes*	
Series	Wiegand26		
Customer code	255		

Cardnumber and customercode at frontend: Numeric (67 / 255)
Remark: Only Wiegand26 is been explained here.

9.15.3. Connections

Signal	Colour	AP1003 connector		AP4x03 connector	Remarks
		Upper	Lower		
Clock	White		12	12	RCP
Data	Green		11	11	RDP
GND	Black		14	14	GND

On Transit PS270 use K4 pin 1 to 3 for connections.
The TRANSIT reader should be featured with wiegand communication (P81 firmware).



9.15.4. Using Combi-booster

If a combi-booster is read by the Transit, first the card number will be sent and after a settable time (preferable 1 sec) the number of the combi-booster. (Only when using the Wiegand Interface.) This combination can be used for the 4-eyes principle.
For making the correct settings, see the Transit Installation Manual and the P81 Firmware Manual

9.15.5. Additional Transit Information

The TRANSIT long range reader and Smartcard-Booster can be applied for applications where smartcards (ISO 14443 and 15693 compatible) are used for building access and also long range driver identification is required. The same access control card can then also be used to facilitate long range vehicle access control. This document describes the integration of this solution into the AEOS access controls system.

Required components;

- TRANSIT reader with P81 firmware (wiegand)
- Smartcard-Booster
- Reader interface: AP1003 or AP4x03

9.15.5.1. CSN or sector information reading

The Smartcard-Booster is featured with an embedded vehicle ID. Moreover the the Smartcard-Booster can read ISO 14443 and 15693 compatible cards. Default Mifare CSN reading is supported. Moreover the unit can also be configured to read sector information by means of the Smartcard-Booster Config software. Free download on http://www.nedapavi.com/downloads/CfgSmartcard-install.zip

When using the Mifare CSN number, we suggest to select the wiegand 37-bit protocol in the TRANSIT reader, see the P81 firmware manual for more information.

9.15.5.2. Setting inter ID delay

The TRANSIT reader will transmit two wiegand messages to the AP1003, first the vehicle ID and after that the driver ID. The delay between these two wiegand messages can be set by means of the P81TEST program over RS232 (default delay = 500 ms).
Free download from http://www.nedapavi.com/downloads/P81Test.zip. Choose ‘Options’ => ‘Dual id control’ from the menu.

9.16. Datalogic DLS2032R-M0 Barcode RS232 interface

Barcode reader Datalogic, consists out of:

- DLS2032R-M0 Barcode slot reader
- CAB-350 RS232Cable 9 PINS female connector 5V
- PG5-05P55 Mains power block

9.16.1. Reader / Interface / Carddata

Reader: DLS2032R-M0 (slot reader)
Interface: RS232
Carddata: Barcode

Remark: Data is depending of Barcode setting in AEOS (Length)
Barcode reader must be programmed to sent the data in RS232 format (read manual and use included Barcode configuration labels)

9.16.2. Firmware and Settings AEOS 2.x

AP1003: AP1003rs232 v1.xx
AP1003rs232 v2.xx
AP4x03: AP4003rs232 v2.xx

AEmon: IdentifierType: Barcode
Start pos = 0, length = 6, Count in digits active
AEserver: IdentifierType: Barcode

9.16.3. Connections

Signal		**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
			Upper	**Lower**		
5	GND	Black	9		9	
2	Tx	Green	10		10	Rx on AEpack

9.17. PIPS License plate recognition – RS232

9.17.1. Reader / Interface / carddata

Reader: License plate camera P372 Spike (Manufacturer: PIPS)
Attention: Depending of the local situation the camera type can be different!
Only Infrared, no overview cam, no wireless communication.
Cable 5 m, bare ended
Reader firmware
Interface: Serial
Card: License plate
Data format:
Remark: License reading is depending of firmware in camera

9.17.2. Software, Firmware and Settings

AEOS: 2.0.5
AEpack: AP1003rs232PIPS_543v200.mhx
AEpack data:

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Alphanumeric		*Format*		
Bit position	0		*Bit position*		
Length	0		*Length*		

Remark: In AEOS at definition of the Identifier type set option *Identifier conversion* to *Uppercase*

9.17.3. Settings License plate camera

See also the documentation of the camera!
Make serial connection to camera

- set internet 172.16.1.2 (IP address camera
- set server 172.16.1.1 (IP address PC with ftp server)
- net broadcast 172.16.255.255 (broadcast address)
- reset

- ftp get nl.eng flash;nl.eng (depending on registration plate)
- at website (172.16.1.2) to configuration -> ANPR engine: change here ANPR engine file to flash;nl.eng
- Again serial:
  - delete flash;uktcc.eng
  - key update
  - reset
  -
- At website logging PIPS op 0x0400040 (for filtering only registration plate)
- On the server make a FTP server, using login: ftp_boot and password: ftp_boot.
  In default ftp map the nl.eng file must be placed.

9.17.4. Connections:

Signal	**Pin**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
RxD (AEpack)	2	10			Receive data
TxD (AEpack)	3	8			Transmit data
GND	5	9			GND

Remark: Power must be supplied external (use large power unit)

9.18. Tattile Vega License plate recognition – RS232

9.18.1. Reader / Interface / carddata

Reader: License plate reader 1024x768 HAD 20fps (Manufacturer: Tattile S.r.l.)
Attention: Depending of the local situation the camera type can be different!
TCP/IP connection for configuration purposes.
Reader firmware
Interface: Serial
Card: License plate
Data format:
Remark: License plate reading is depending of firmware in camera.
Dutch string is e.g. : 12DDRR-NLD

9.18.2. Software, Firmware and Settings

AEOS: 2.1
AEpack: AP1003rs232alpha
AEpack data: 010B xxxxxx

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Alphanumeric		*Format*		
Bit position	0		*Bit position*		
Length	6 (or 0, see below)		*Length*		

Remark: In AEOS at definition of the Identifier type set option *Identifier conversion* to *Uppercase*

9.18.3. Settings license plate camera

Check the documentation of the camera and try first to connect e.g. hyperterminal to check the string coming from the camera

With camera versions up from 1-2008 the output string can be regulated. Only the license number will be sent followed with CR / LF. To achieve this make following settings at the camera:
Menu*:* *Event/action, at OSC READ- RS232:*
Enable: yes
Message: %PLATE_STRING%0x0D%0x0A
System / serial protocol settings
Message: RAW
! At the generic identifier the Badge number length must now be set to 0.

9.18.4. Connections:

Signal	**Pin**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
RxD (AEpack)	2	10			Receive data
TxD (AEpack)	3	8			Transmit data
GND	5	9			GND

9.19. Tattile Vega License plate recognition – IP

9.19.1. Reader / Interface / carddata

Reader: License plate reader 1024x768 HAD 20fps (Manufacturer: Tattile S.r.l.)
Attention: Depending of the local situation the camera type can be different!
TCP/IP connection for configuration purposes.
Reader firmware
Interface: IP
Card: License plate
Data format:
Remark: License plate reading is depending of firmware in camera.
Dutch string is e.g. : 12DDRR-NLD ??

9.19.2. Software, Firmware and Settings

AEOS: 2.3.x
AEpack: Not needed
AEpack data:

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Alphanumeric		*Format*		
Bit position	0		*Bit position*		
Length	0		*Length*		

Remark: In AEOS at definition of the Identifier type set option *Identifier conversion* to *Uppercase*

9.19.3. Settings license plate camera

The camera should be configured like this:
Click the link “Event Actions” and then the item in the table in the row “OCR Read” and the column “TCP Msg”. Then edit the settings according to this screen:

The Server IP should be the IP of the AEpu where the IPBadge AEbc is deployed.
The Server Port should be equal to the port configured in the IPBadge AEbc.

9.20. Sony XCI-NPR License plate recognition – RS232

9.20.1. Reader / Interface / carddata

Reader: License plate reader XCI-NPR (Manufacturer: Sony)
Reader firmware
Interface: Serial
Card: License plate
Data format:

9.20.2. Software, Firmware and Settings

AEOS: 2.2.5
AEpack: AP1003rs232alpha
AEpack data: 010B 534248523031 License number: SB HR 01

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Alphanumeric		*Format*		
Bit position	0		*Bit position*		
Length	0		*Length*		

Remark: In AEOS at definition of the Identifier type set option *Identifier conversion* to *Uppercase*

9.20.3. Connections:

Set Camera to RS232

Signal	**Pin**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
RxD (AEpack)	6	10			Receive data
TxD (AEpack)	7	8			Transmit data
GND	8	9			GND

9.21. Nedap ANPR License plate recognition – IP

Settings are identical to Tatile cameras, see chapter 9.19.

So at IPBadge AEbc the *Tatile Car plate* must be selected.

9.22. Mobile Phone MC35i – RS232 interface

The MC35i acts as receiver for telephone and sends phone number serial to AEpack. Phone number is used for autorisation data.

9.22.1. Reader / Interface / carddata

Reader : Siemens MC35i
The MC35i Terminal is a robust unit with integrated GPRS and Dual-Band GSM functionality.
Reader firmware :
Interface : Serial RS232 (fixed settings?)
Card : Telephone number
Data format : 10 bytes alpha numeric (phone number, starting with +)

Remark Modem rings only once and will directly go to busy (is communication with AEpack is available). This results that the user will only hear the busy ('hook on') tone.

9.22.2. Software, Firmware and Settings

AEOS Version 2.1build88
AEpack :AP1003rs232MC35i_543v200
AP4003rs232MC35i_543v200

AEpack data: 010C 2B3331353434343731363835 card number: +31544471685

Generic: **AEmon**: IdentifierType: Generic (default settings)

Badge number			**Customer code**		
Format	Ascii Alphanumeric		*Format*		
Bit position			*Bit position*		
Length			*Length*		

AEserver: IdentifierType: Generic, settings see AEmon

Printable		*Print receipt*	
Identifier length			
Identification Type	Alphanumeric		
Identifier min. value		*Identifier max. value*	
Has different intern.		*Add leading zeroes*	
Customer code length			
Customer code			
Subtype		*Customer code type*	

9.22.3. Connections

Pin 9 Pole SUB-D	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
1					--
2		10		10	Receive data from MC35i
3		8		8	Transmit data to MC35i
4					--
5		9		9	Gnd

Settings: Baudrate 9600 Bd
Data settings 8 databits, No Parity, 1 Stopbit

9.23. Storm PIN pad – RS232 interface

The Storm PIN pad terminal is used for entering the identifier number, ending with an #. The received data (as ASCII) is used as identifier number.
Can be used as identifier and as PIN code terminal (check correct firmware)

9.23.1. Reader / Interface / carddata

Reader : Storm PIN code terminal, Nedap art. Nr 9896090
The Storm PIN code terminal sends Ascii Alphanumeric data.
Reader firmware :
Interface : Serial RS232
Card : Entered code on PIN pad, max 24 characters
Data format : Alphanumeric

Remark Entered PIN nr is used as identifier. # is used as terminator. If 5 seconds no data (PIN) is received, the buffer is cleared.

9.23.2. Software, Firmware and Settings for Identification

AEOS Version 2.1.7
AEpack :AP1003rs232Storm_543v200 (not for PIN code verification)
AP4003: Not available

AEpack data: 010B31323334 cardnumber: 1234 (entered PIN code: 1234#)

Generic: **AEmon**: IdentifierType: Generic (default settings)

Badge number			**Customer code**		
Format	ASCII-Hexadecimal	Hex	*Format*		
Bit position			*Bit position*		
Length			*Length*		

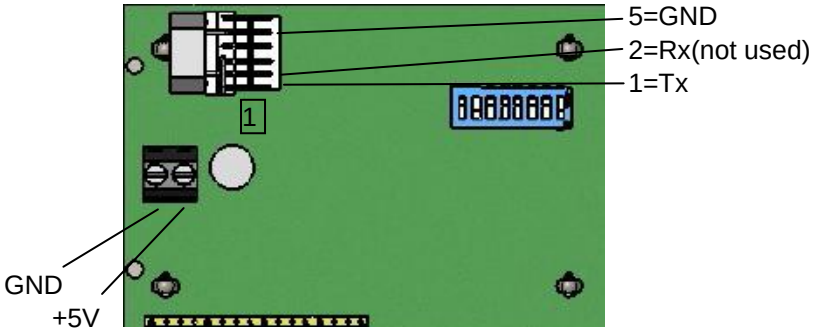
All data entered is read (length 0 means all data). Of course you can limit the amount of characters if needed. (Setting above could also work with ASCII Alphanumeric instead of ASCII Hexadecimal)

9.23.3. Connections

Pin 9 Pole SUB-D	**PIN Pad**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
2	2	10	10 *		Transmit data from PIN pad
3	1	8			Receive data for PIN pad (not used)
5	5	9	9 *		Gnd

*: For using PIN code at AP1003 use Lower connector, for identification use Upper connector.

Settings: DIP Switches: 1 and 7: ON, others OFF
9600 BD, No Parity, 8 databits, 1 stopbit
Power: +5V, GND using external Power Supply



9.24. GEMPlus GEM Easy Access300 - Mifare

Reads Mifare CSN number, Wiegand32 bin output

9.24.1. Reader / Interface / carddata

Reader : GemPlus GemEasyAccess300
Reader firmware :
Interface : Wiegand 32bin
Card : Mifare
Data format : CSN number

Remark CSN number is sent over Wiegand32bin. In example below the card number must be represented decimal, so Generic Identifier is used.
(If Hexadecimal data can be used , Wiegand32Binairy can be selected at the frontend)

9.24.2. Software, Firmware and Settings

AEOS Version 2.2.3
AEpack : AP1003wieg32binRev_543v208
AP4003wieg32binRev_543v207

AEpack data: 0304C45C82DD card number: 3294397149 (decimal)

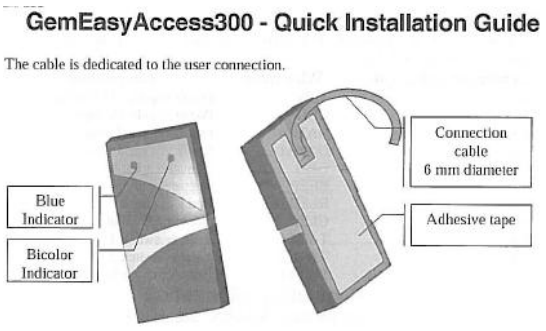
Generic: **AEmon**: IdentifierType: Generic (default settings)

Badge number			**Customer code**		
Format	Hexadecimal	Numeric	*Format*		
Bit position	0		*Bit position*		
Length	32		*Length*		

AEserver: IdentifierType: Generic, settings see AEmon

9.24.3. GemEasyAccess300 Connections

Pin	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
Data-0	Green		12	12	Data-0
Data-1	White		11	11	Data-1
GLed	Grey	14		13	Led Green
RLed	Purple	15		15	Led Red
GND	Black		14	14	GND
+12V	Red	12 (12V)		1 (12V)	+12V (AP4x03 attention!)
Wiegand GND	Blue	9		9	Ground Wiegand


GemEasyAccess300 - Quick Installation Guide
The cable is dedicated to the user connection.
Blue Indicator
Bicolor Indicator
Connection cable 6 mm diameter
Adhesive tape

GemEasyAccess300 Connections

Shielded AWG28 10-wire cables are used to connect the device with the corresponding functions as illustrated below:

Conductor Cable Color	Wire Name	Designation
Red	+12V	Power supply +12V input
Black	GND	Power supply 0V input
Green	DATA0	Data0 Wiegand output
White	DATA1	Data1 Wiegand output
Blue	GND	Wiegand ground
Pink	BEEP	Beeper input
Purple	RLED	Red LED input
Grey	GLED	Green LED input
Orange	TAMP1	Tamper switch 1
Yellow	TAMP2	Tamper switch 2
	SHIELD	Connect to earth ground

9.25. Dorset LID650 (Trovan)

Reads Trovan serial number of cards, ASCII output. (www.dorset.nu)

9.25.1. Reader / Interface / carddata

Reader : Dorset LID650
Reader firmware : v724
Interface : RS232 ASCII
Card : Trovan
Data format : Card serial number

Remark Card serial number is sent over RS232, in ASCII Hexadecimal format.
Using the generic identifier will transfer this to decimal representation.

9.25.2. Software, Firmware and Settings

AEOS Version 2.2.3
AEpack AP1003rs232alpha_543v206
AP4003rs232alpha_543v203

AEpack data: 010B 30303036383545423633 = 000685EB63 (hex) --> card nr 109439843 (dec.)

Generic: **AEmon**: IdentifierType: Generic

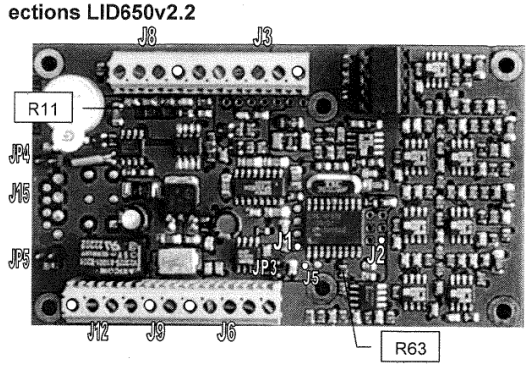
Badge number			**Customer code**		
Format	ASCII-Hexadecimal	Numeric	*Format*		
Bit position	0		*Bit position*		
Length	10		*Length*		

AEserver: IdentifierType: Generic, settings see AEmon

9.25.3. Dorset LID 650 Connections

RS232 or RS485 can be used, default 9600, N, 8, 1.
Power for reader is not supplied by AEpack

Pin	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
J6 – 1		10		10	Receive data RS232
J6 – 3		13		8	RS485 B
J6 – 4		16		10	RS485 A
J6 – 5			9	9	GND



J6 Communication port

1. Txd, transmit signal RS232 serial port (standard 9600,n,8,1)*
2. Rxd, receive signal RS232 serial port (standard 9600,n,8,1)*
3. RS485 B (19200,n,8,1*) or optional TTL output for wiegand**
4. RS485 A (19200,n,8,1*) or optional TTL output for wiegand**
5. Gnd, Ground interface

Important note: Depending on firmware version V6.xx or N6.xx RS232 or RS485 is available.

9.26. IS3480 QuantumE (Metrologic) - Barcode

Reads Barcode labels.

9.26.1. Reader / Interface / carddata

Reader : Metrologic IS3480 QuantumE
Interface : RS232 ASCII
Card : Barcode

Remark Barcode number is sent over RS232.
With Generic identifier it is transferred to Barcode identifier (is one of the possibilities) (using the *Target Identifier* in the Generic settings at AEmon).

9.26.2. Software, Firmware and Settings

AEOS Version 2.3.7
AEpack AP1003rs232_543v20x
AP4003rs232_543v201

AEpack data: unknown

Generic: **AEmon**: IdentifierType: Generic

Badge number			**Customer code**		
Format	BCD		*Format*		
Bit position	0		*Bit position*		
Length	0		*Length*		
	Target Identifier: Barcode				

AEserver: IdentifierType: Barcode

9.26.3. Connections

Pin 9 Pole SUB-D		**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
2		10	*	10	Transmit data from Barcode reader
3		8		8	Receive data (not used)
5		9		9	GND

9.27. Convexs / Invexs readers (Nedap Mifare sector data NR0077 / NR0125)

For other usage see also the Convexs and Invexs user guide (www.nedap.net) for more information. Here only the NR0077 and NR0125 are explained.

9.27.1. Reader / Interface / card data

Reader : Nedap Readers (Convexs, Invexs)
Configuration : NR0077 or NR1025 (differs in security keys)
Interface : RS485, RFdata (depends on used configuration in Nedap reader)
Card : Mifare, sector data with customer code and card number
(*MF0010 0000123456)*

9.27.2. Software, Firmware and Settings

AEOS Version 2.3.8
AEpack AP1003rs485NR / AP1003rs485NR2
AP4003rs485NR / AP4003rs485NR2
AP6003rs485NR / AP6003rs485NR2

AEpack data: 0502 4d4630303130 0000123456

Generic: **AEmon**: IdentifierType: Generic

Badge number			**Customer code**		
Format	BCD		*Format*	ASCII_Alphanummeric	
Bit position	48		*Bit position*	0	
Length	10 digits		*Length*	6 bytes	

AEserver: IdentifierType: Generic, customer code MFxxxx (**must** be filled in)

9.27.3. Connections

Convexs / Invexs		**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
A1		1 or 12		1	Power (10-35VDC)
A2		9		9	GND
A3		13		8	A (-)
A4		16		10	B (+)

See also installation sheet Convexs / Invexs.

9.28. IButton RS232

End user : Nedap AVI (Netherlands)

Date	5-2006	*Contact Groenlo*	MKO
Country	Netherlands	*Business Partner*	Nedap AVI
		Contact BP	

9.28.1. Reader / Interface / carddata

Reader: IButton
Converter box (no manufacturer data), DB25+hardware included (no manufacturer data)
Reader firmware
Interface: RS232
Card: IButton (type DS 1990A-f3)
Data format: Alpha-numeric (use e.g. Hyperterminal to test)

9.28.2. Software, Firmware and Settings

AEOS: 2.0.5.1
AEpack: AP1003rs232alpha_543v204.mhx
AP4003rs232alpha_543v200.mhx

AEpack data: 010B 45313031364143424543303930303030304236

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Ascii-Alphanumeric		*Format*		
Bit position	0		*Bit position*		
Length	0 bits		*Length*		

Remark: Depending of the part of data that is unique for this customer above settings can be changed to use only a part of the complete card number.
Customer code: See Remark above

AEserver: IdentifierType: Generic, settings see AEmon

Printable		*Print receipt*	
Identifier length	0		
Identification Type	Alpha numeric		
Identifier min. value		*Identifier max. value*	
Has different intern.		*Add leading zeroes*	
Customer code length	0		
Customer code			
Subtype	1	*Customer code type*	

9.28.3. Connections

Signal	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
DB25-GND (7)		9		9	GND
DB25-Power (20)		11		1	+5V
DB25-Tx (3)		10		10	Rx

9.29. MOTOROLA – FlexPass FP3515A (Omron)

Date	12/2009	*Contact Groenlo*	R. Waenink
Country	Germany	*Business Partner*	Nedap Duitsland
		Contact BP	A. Bouter

9.29.1. Reader / Interface / carddata

Reader : Motorola FlexPass FP3515A / 11037
Reader firmware :
Interface : Omron, ABA Track 2 OC
Card : Indala
Data format : Omron

9.29.2. Software, Firmware and Settings

AEOS Version 2.3.9
AEpack : APx003omron
AEpack data: 020200 00224 32848 Carddata: 21049478-1 32848
00 is no part of badge number
Customercode: 00224
Card nummer: 32848

AEmon: IdentifierType: Omron (or use: Generic Identifier)
Badge ID: starts at 5, number of digits 5
Customer ID: starts at 0, number of digits: 5
AEserver: IdentifierType: Omron
Identification Type: BCD, Identifier length: 5
Customer code: 000224 (6 positions!)

9.29.3. Connections

Pin	Colour	AP1003 connector		AP4x03 connector	Remarks
		Upper	Lower		
1	Black	9		14	0V (GND)
2	Red	11		1 *	+5V * Check power AP4x003 !
4	Green		11	11	Data
5	White		12	12	Clock
7				13	Led R (NA)
8				15	Led V (UL)

9.30. Satel CZ-EMM (EM)

Date	12/2009	*Contact Groenlo*	M.Krabben
Country	Netherlands	*Business Partner*	ALTEST Sp. z o.o. Poland
		Contact BP	Ożga Błażej

9.30.1. Reader / Interface / carddata

Reader : CZ-EMM2 / CZ-EMM3
Reader firmware : default
Interface : Wiegand
Card : Satel default
Data format : EM

Remark: Reader uses only one data connection. Data is sent as raw data on this connection.

9.30.2. Software, Firmware and Settings

AEOS Version 2.3.9
AEpack AP1003EMcode_543_v200

AEpack data: `0901 2300f66255 cardnr: 0150340002389`

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Hexadecimal	Numeric	*Format*	Ascii-Alphanumeric	
Bit position	0		*Bit position*	0	
Length	40		*Length*	0	

AEserver: IdentifierType: Generic, settings see AEmon

9.30.3. Connections

Pin	**AP1003 connector**		**Remarks**
	Upper	**Lower**	
Red	12		Vcc (12VDC)
Blue	9		GND
Green		11	Data IN
Brown	(12)		Not connected / to Vcc (pin 12) for EMM3
Grey		15	NA
Pink		13	UL
Yellow			Buzzer
White			Input reader sabotage (1K resistor)

Remark: CZ-EMM3 reader is identical to CZ-EMM2, with difference that this reader **must** be enabled by connecting brown wire to Vcc (Pin 12)
Customer: BreBank Polen

9.31. PhG readers (e.g. Oris, Voxio)

PhG readers can be used in case e.g. Legic cards must be read. PhG readers are available in different versions, the firmware at these readers determine their functionality. Always carefully check the documentation included with the reader about the connections.
The used protocol and connections are determined by the PhG reader, not by AEOS. Depending on the eventually existing readers at a customer the identifier must be adjusted to these existing readers.

There are several options:

- Bus communication using RS485 with the PhG Crypt protocol, available for the PhG readers with this protocol.
  The PhG crypt protocol is available up from AEOS 2.3.13 and only available for:
  - AP1003 Max 8 PhG Crypt readers can be connected
  - AP4x03 Max 4 PhG Crypt readers can be connected

  For these readers PIN code is possible (only 4 digits).
  Settings what data to be read and what to be sent is determined by the settings made in PhG reader (these settings are **not** made by the Nedap software).
- Point to Point communication (Using one of the existing protocols as Omron, Wiegand, Serial, etc)
  Check is the available firmware for the AEpacks matches the protocol of the used PhG reader.
  Settings what data to be read and what to be sent is determined by the PhG reader.
- Custom made specials (e.g. PhGCrypts76) are not described here.

Below some examples how PhG readers can be connected to AEOS

9.32. PhG Crypt bus protocol

For applying the PhG Crypt bus protocol, the (free) BusDevice-Integrator AEbc must be used. This AEbc controls the communication with the PhG Crypt bus. This AEbc expands detected PhG readers to virtual devices for AEmon. On these virtual devices the correct bindings can then be made (*Reader, Pin, UL, NA, Ye and Buzzer*).
For detailed information see the *AEbc Integrator Descriptions*, chapter *BusDevice-Integrator*.
LED's and buzzer are controlled by the bus communication.

9.32.1. PhG Crypt Reader / Interface / carddata

Reader : PHG readers using PhG Crypt
Reader firmware :
Interface : RS485 PhG Crypt
Card : Depending of PhG
Data format : Depending of used card technique

9.32.2. PhG Crypt Software, Firmware and Settings

AEOS : Version 2.3.13
AEpack : AP1003rs485PHGCrypt_543v2.xx
: AP4003rs485PHGCrypt_543v2.xx
AEpack data (e.g.): `0000 1483648360` Card number is 1483648360
The AEpack data is depending on how the PhG readers are configured.

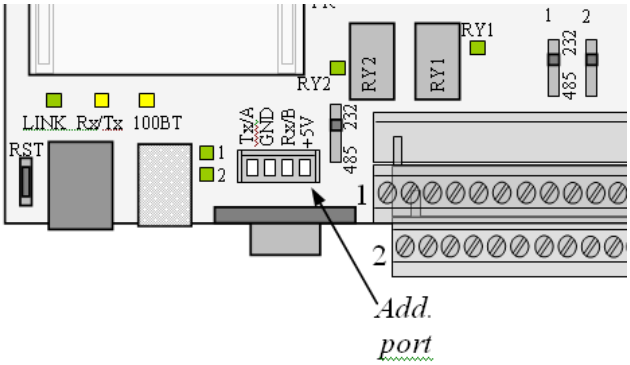
AEmon: IdentifierType: Generic
Depending on the used cards. There is no general setting available.
AEserver: IdentifierType: See settings at AEmon.

9.32.3. PhG Crypt Connections

PhG		AP1003 connector		AP4x03X connector 4 PIN	Remarks
Name	**PIN**	**Upper**	**Lower**		
B (RS485)	3	13		3	RS485 B (+)
A (RS485)	4	16		1	RS485 A (-)
-Ub (GND)	6		1	2	GND
+Ub	7	1 (12V ?)			Power (for all connected readers)

Remarks

- The power must be taken from a suitable power supply. This can be from the used AEpacks, but it must be sufficient to feed all connected readers.
- For the AP4003 the PhG readers are connected into a bus, not point to point. Inputs and Outputs are connected to the (normal) peripheral connectors.
- For the AP4003 the connection with the PhG readers is by the small additional connector (*Add. Port,* see below) and **NOT** through the normal reader connectors.
  The belonging switch must be set to 485.



9.33. PhG Oris 100 Omron

9.33.1. PhG Oris 100 Omron Reader / Interface / carddata

Reader : PHG Oris 100
Reader firmware : Depending on customer
Interface : Omron
Card : Legic
Data format : Omron 10 bytes

9.33.2. PhG Oris 100 Omron Software, Firmware and Settings

AEOS : Version 2.2 build 63
AEpack : AP1003omron_543v204
: AP4003omron_543v202
AEpack data: `020200 1483648360` Card number is 1483648360
00 is no part of badge number

AEmon: IdentifierType: Omron

Omron settings	
Badge ID, starts at digit position	0
Badge ID, number of digits	**10**
Customer ID, starts at position	0
Customer ID, number of digits	0

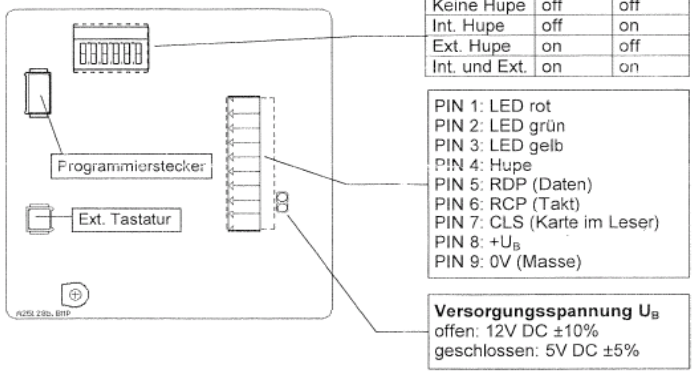
AEserver: IdentifierType: Omron

Printable		*Print receipt*	
Identifier length	10		
Identification Type	BCD		
Identifier min. value	1	*Identifier max. value*	100000000000
Has different intern.		*Add leading zeroes*	
Customer code length			
Customer code			
Subtype		*Customer code type*	

9.33.3. PhG Oris 100 Omron Connections

Pin	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
7	Blue		16		CLS
5	Black		11	11	RCP
6	White		12	12	RDP
2	Green	14		13	Led Green
1	Black	15		15	Led Red
9	Black		14	14	GND
8	Red	12 (12V)		1 (5V)	+5V (50mA ?)

Anschlussplan ORIS100 mit Magnetleser-Schnittstelle, Artikel-Nr. 13502



Funktion	DIP 4	DIP 5
Keine Hupe	off	off
Int. Hupe	off	on
Ext. Hupe	on	off
Int. und Ext.	on	on

DIP Switch:
1 ON (Yellow LED)
2 OFF
3 OFF
4 ON
5 ON
6 OFF

05.04.2004 / A13502_69549.DOC

9.34. PhG Aperio 200 Omron

In this example no link is available between the card number read and the card number printed on the card.

9.34.1. PhG Aperio 200 Omron Reader / Interface / carddata

Reader : PHG nr 10593 (HW: 32689c)
Reader firmware : Software 69679f.hex (Hitag1-SerNr_BCD.hit)
Interface : Omron (10 bytes)
Card : Hitag
Data format : 10 bytes BCD

9.34.2. PhG Aperio 200 Omron Software, Firmware and Settings

AEOS Version 2.0.7
AEpack : AP1003omron_543v204
: AP4003omron_543v202

AEpack data: `020200 0492276069` card number: 600 0653:
00 is no part of badge number
There is no link between card number and data.

Generic: **AEmon**: IdentifierType: Generic

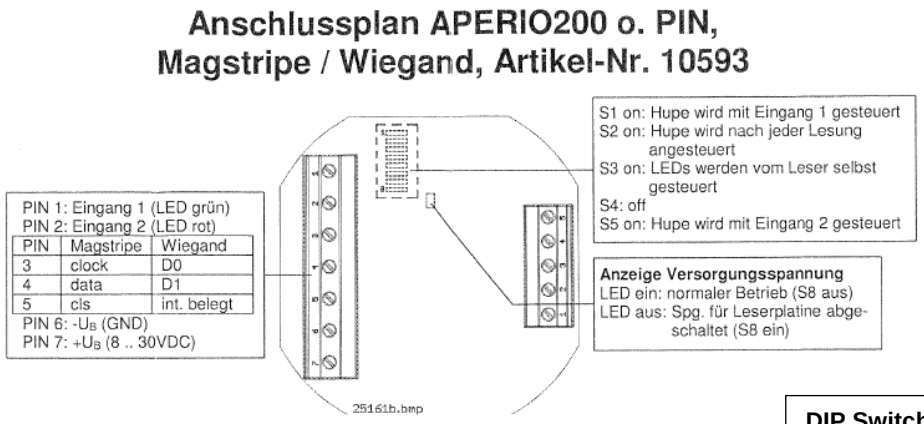
Badge number			**Customer code**		
Format	BCD		*Format*		
Bit position	8		*Bit position*		
Length			*Length*		

AEserver: IdentifierType: Generic, settings see AEmon

Printable		*Print receipt*	
Identifier length			
Identification Type	BCD		
Identifier min. value		*Identifier max. value*	
Has different intern.		*Add leading zeroes*	
Customer code length			
Customer code			
Subtype		*Customer code type*	

9.34.3. PhG Aperio 200 Omron Connections

Pin	Colour	AP1003 connector		AP4x03 connector	Remarks
		Upper	Lower		
1		14		13	LED Green
2		15		15	LED Red
3			12	12	Data 0 (RCP)
4			11	11	Data 1 (RDP)
5					-- Not used
6			14	14	-Ub (GND)
7		12		External	+Ub (Vcc 8 –30VDC)


Anschlussplan APERIO200 o. PIN,
Magstripe / Wiegand, Artikel-Nr. 10593
S1 on: Hupe wird mit Eingang 1 gesteuert
S2 on: Hupe wird nach jeder Lesung angesteuert
S3 on: LEDs werden vom Leser selbst gesteuert
S4: off
S5 on: Hupe wird mit Eingang 2 gesteuert
Anzeige Versorgungsspannung
LED ein: normaler Betrieb (S8 aus)
LED aus: Spg. für Leserplatine abgeschaltet (S8 ein)
PIN 1: Eingang 1 (LED grün)
PIN 2: Eingang 2 (LED rot)
PIN Magstripe Wiegand
3 clock D0
4 data D1
5 cls int. belegt
PIN 6: -UB (GND)
PIN 7: +UB (8 .. 30VDC)
25161b.bmp

DIP Switch:
1 FF
2 ON (LED control external)
3-8 OFF

9.35. PhG Voxio RS485

Reader is sending the card number as plain text.

9.35.1. PhG Voxio RS485 Reader / Interface / carddata

Reader : Voxio
Interface : Serieel RS485
Card:
Data format : ASCII

Remark: Readers are equipped with PIN Pad, this PIN Pad cannot be used (not implemented)

9.35.2. PhG Voxio RS485 Software, Firmware and Settings

AEOS Version 2.1.2
AEpack AP1003rs232alpha_543v205
AP4003rs232alpha_543v20x
AEpack data: `010B 373130393930` Cardnr: 990 (first bytes are in this example not used)

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Ascii-Hexadecimal	Hex	*Format*	Ascii-Alphanumeric	
Bit position	16		*Bit position*		
Length	4		*Length*		

AEserver: IdentifierType: Generic, settings see AEmon

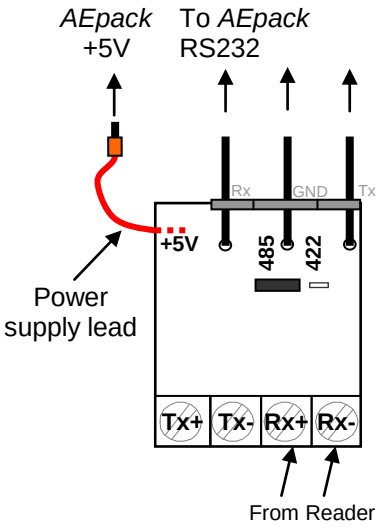
9.35.3. PhG Voxio RS485 Connections

Pin	**AP1003 connector**		**AP4x03 connector**	**Remarks**
	Upper	**Lower**		
4	13		8	A
3	16		9	B
7	12			12V
6		14	14	GND
1	14		13	Led Green
2	15		15	Led Red

AP1003 up from PMS D have RS485 included, AP4x03X can be switched between RS232 and RS485.

9.35.4. PhG Voxio RS485 Connections with AX1012

Pin	**AP1003 connector**		**AP4x03 connector**	**AX1012**	**Remarks**
	Upper	**Lower**			
4				3 (Rx+)	
3				4 (Rx-)	
	11			Red (5V)	
7	12				12V
6		14	14		GND
1	14		13		Led Green
2	15		15		Led Red



9.36. PhG Voxio 200 Wiegand

In this example no link is available between the card number read and the card number printed on the card. Depending on the PhG protocol use the appropriate firmware for the AEpack.

9.36.1. PhG Voxio 200 Wiegand Reader / Interface / carddata

Reader : Voxio 200
HW: 32689e SW: 69679j.hex
Interface : Wiegand Universal MDB first (Hitag1_CSN_WUniversal.hit)
Card: Hitag1
Data format : Wiegand / Hex

Remark: Data on card is from sector, Nedap uses card serial number.
02CHR 4217 FB150878
02CHR 4211 8F180878
00BHR 7350 2F290878
Settings for PHG reader are special made: use file: Hitag1_CSN_WUniversal.hit).

9.36.2. PhG Voxio 200 Wiegand Software, Firmware and Settings

AEOS Version 2.2.3 build17
AEpack AP1003wieg128max
AP4003wieg128max
AEpack data: `0308 30 00008F180878` Cardnr: 4211 (8F180878 in AEOS)

AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	Hexadecimal	Hex	*Format*	Ascii-Alphanumeric	
Bit position	24		*Bit position*		
Length	0		*Length*		

AEserver: IdentifierType: Generic, settings see AEmon
Cardnummer 1 – ffffffffffff (12) ?

9.36.3. PhG Voxio 200 Wiegand Connections

Pin	**Colour**	**AP1003 connector**		**AP4x03 connector**	**Remarks**
		Upper	**Lower**		
4			11	11	RCP
3			12	12	RDP
1		14		13	Led Green
2		15		15	Led Red
6			14	14	GND
7		12 (12V)		1 (12V)	+V (8..30VDC)

9.37. GigaTMS MF7 (Mifare)

Date	06/2010	*Contact Groenlo*	H. Braskamp
Country	Netherlands	*Business Partner*	Nedap Groenlo, Retail / Cube
		Contact BP	Jeroen Struycken

9.37.1. Reader / Interface / carddata

Reader : GigaTMS MF7
Reader firmware : ?
Interface : RS485
Card : Mifare
Data format : CSN

Remark: Data and LED's are controlled over RS485. Mifare CSN sent as Nedap format (0501) in identical format as the Convexs readers.
Reader used at Nedap

9.37.2. Software, Firmware and Settings

AEOS Version 2.3.12
AEpack AP1003rs485GigaTMSMF7
AP4003rs485GigaTMSMF7

AEpack data: `0501 7f4D5739 Cardnr CSN: 7f4D5739`

AEmon: IdentifierType: Mifare
AEserver: IdentifierType: Mifare CSN

9.37.3. Connections

Pin	**AP1003 connector**		**AP4x03 connector**	**Remarks**
	Upper	**Lower**		
Yellow	13		8	RS485 A(+)
Blue	16		10	RS485 (B-)
Black - shield	6		14	GND
Red	12		1	Vcc (5 –18VDC) (check pin 1 on AP4003)

Led connections are not used, the LED's are controlled by the RS485 communication.

9.38. Symbol LS9208 barcode reader (Barcode)

Date	06/2010	*Contact Groenlo*	R.Waenink
Country	Netherlands	*Business Partner*	Nedap Poland
		Contact BP	

9.38.1. Reader / Interface / carddata

Reader : Symbol LS9208
Reader firmware : ?
Interface : RS232
Card : Barcode
Data format : Barcode

9.38.2. Software, Firmware and Settings

AEOS Version 2.3.13
AEpack AP1003rs232_543v203
AP4003rs232_543v201

AEpack data:
```
0002 02062980025181          Cardnr: 2062980025181
0002 8041101000000001011000  Cardnr: 8041101000000001011000
```

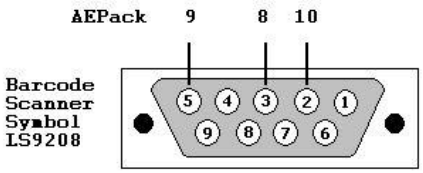
AEmon: IdentifierType: Generic

Badge number			**Customer code**		
Format	BCD		*Format*	Ascii-Alphanumeric	
Bit position	0		*Bit position*		
Length	0		*Length*		

AEserver: IdentifierType: Generic, settings see AEmon
Identifier length is set to 0, due to the different length of the offered barcodes.

9.38.3. Connections

Pinnr 9 PIN SUB D	**AP1003 connector**		**AP4x03 connector**	**Remarks**
	Upper	**Lower**		
3	8		8	Tx (RS232)
5	9		9	GND
2	10		10	Rx (RS232)



Remark: The LS9208 first must be correct configured for sending RS-232, Enabling interleaved 2 of 5, I 2 of 5 – Any length. The correct barcodes for these settings can be found in the LS9208 manual (and below)

9.39. Evis EDI-05 RS485 (Legic, Mifare)

Date	01/2011	*Contact Groenlo*	R. Waenink
Country	Netherlands	*Business Partner*	Tyco CH
		Contact BP	

9.39.1. Reader / Interface / carddata

Reader : Evis EDI-05
Reader firmware : ?
Interface : RS485
Card : Card serial nr, block data , Legic, Mifare, etc, depending on reader configuration
Data format : ASCII

Remark: Data and LED's are controlled over RS485
PIN is used over RS485 (select *PinPort* in Verification AEbc)

9.39.2. Software, Firmware and Settings

AEOS Version 2.4.4
AEpack AP1003rs485EVIS_543
AP4003rs485EVIS_543
AP6003rs485EVIS_543

AEpack data:
```
0110000000000479691  Cardnr Blockdata: 479691
0110000000000479692  Cardnr Blockdata: 479692
0110000001092400591  Cardnr CSN: 1092400591
0110000001092400671  Cardnr CSN: 1092400671
```

AEmon: IdentifierType: Generic (as example)

Badge number			**Customer code**		
Format	BCD		*Format*	Ascii-Alphanumeric	
Bit position	0		*Bit position*		
Length	16 digits		*Length*		

AEserver: IdentifierType: Generic, settings see AEmon

9.39.3. Connections

Pin	**AP1003 connector**		**AP4x03 connector**	**AP6003 connector**	**Remarks**
	Upper	**Lower**			
X1 – 6	13		8	8	RS485 A(+)
X1 – 5	16		10	10	RS485 (B-)
X2 – 1	6		14	6	GND
X2 – 2	1 (12)		1	1	Vcc (12VDC) (check pin 1 on AP4003) **

**: Remark: AP4x03 reader power supply:
Reader at 24V: only 1 reader to be connected due to startup peak current of Evis reader.
Reader at 12V: 2 readers can be connected (due to startup peak current of Evis reader)
If more than 2 readers give problems, use external power supply for feeding the readers.

Remark: Due to 2 way communication, Evis reader will not always respond when card is read at the same time. This can respond in no reading of the card when e.g. a LED is controlled. The user has to offer his badge again to the reader (this flaw has seen more at the AP6003 than at the AP1003 / AP4003)

Remark: LEDs and buzzer (AP6003) are controlled by the RS485 communication.
DIP switches (S1) all set to 0

10. DOCUMENT HISTORY

Date	Nr	Who	Changes
02-11-2012	23	REW	Nedap APNR license plate camera added
16-01-2012	23	REW	Evis EDI-05 (optional with PIN) added
07-12-2010	22	REW	Mifare Plus CSN4 and Mifare Plus CSN7 removed
02-11-2010	21	REW	Generic ASCII Num. added, Omron 00 bytes declared
14-10-2010	20	REW	Storm PIN pad identifier setting changed to ASCII Hexadecimal
30-06-2010	19	REW	PhG Crypt, Barcode LS9208 added
17-06-2010	18	REW	GigaTMS MF7 added
26-04-2010	17	REW	Wiegand36 added
23-03-2010	16	REW	PIN code leading zero's added
22-02-2010	15	REW	PhG readers as examples added
11-12-2009	14	REW	Omron200max added
04-12-2009	13	REW	Wieg128MaxMultiPin, Wieg26C1000 added, verification updated
19-11-2009	12	REW	Nedap readers (Convexs, Invexs) NR0077 (chapter 9.27)added
12-11-2009	11	REW	Metrologic barcode scanner (chapter 9.26)added
08-09-2009	10	REW	Identifier list (chapter2.8) expanded
06-04-2009	9	REW	Added AP6003, Generic Target Identifier, Trovan and Tattili over IP